KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TEK KAMERA KULLANARAK NESNE-KAMERA ARASINDAKİ UZAKLIĞIN BELİRLENMESİ İÇİN BİR YÖNTEM

FATMA KUNCAN

KOCAELİ 2015

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TEK KAMERA KULLANARAK NESNE-KAMERA ARASINDAKİ UZAKLIĞIN BELİRLENMESİ İÇİN BİR YÖNTEM

FATMA KUNCAN

Doç. Dr. Mehmet YILDIRIM
Danışman, Kocaeli Üniv. .....

Yrd. Doç. Dr. Halil YİĞİT
Jüri Üyesi, Kocaeli Üniv. .....

Yrd. Doç. Dr. Adem TUNCER
Jüri Üyesi, Yalova Üniv. .....

Tezin Savunulduğu Tarih: 28.05.2015

ÖNSÖZ ve TEŞEKKÜR

Bu çalışmada, son yıllardaki teknolojik gelişimlere bağlı olarak, birçok alanda yaygın bir şekilde kullanılan görüntü işleme metotları kullanılarak, nesnenin kameraya olan uzaklığı belirlenmiştir. Bu çalışmada, nesnenin kameraya olan uzaklığını ölçmek için, tek kameranın kullanıldığı bir yöntem önerilmiştir.

Yüksek lisans tez çalışmam boyunca, akademik anlamda gösterdiği destek ve çalışmalarım süresince gösterdiği yakın ilgisi için değerli danışmanım ve hocam Sn. Doç. Dr. Mehmet Yıldırım'a teşekkür ederim.

Ayrıca hayatım boyunca desteklerini bir an olsun esirgemeyen ve beni bugünlere getiren aileme sonsuz minnet duygularımı sunarım. Son olarak desteğini bir an olsun eksik etmeyen eşim ve oğluma sonsuz minnet duygularımı sunarım. Onların destekleri ve duaları olmadan hayatta başarılı olabilmem imkansız olurdu.

Mayıs – 2015 Fatma KUNCAN

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iv
TABLOLAR DİZİNİ vi
SİMGELER DİZİNİ VE KISALTMALAR vii
ÖZET viii
ABSTRACT ix
GİRİŞ 1
1. GÖRÜNTÜ İŞLEME VE NESNE TESPİTİNDE KULLANILAN YÖNTEMLER 3
1.1. Renk Tabanlı Yaklaşımlar 4
1.2. Renk Modelleri 5
1.2.1. RGB renk modeli 7
1.2.2. CMY/CMYK renk modeli 8
1.2.3. HSV renk modeli 9
1.3. Nesne Tespitinde Kullanılan Görüntü İşleme Yöntemleri 10
1.3.1. Kullanılan görüntü işleme uygulamaları 10
1.3.2. RGB den HSV ye aktarılma 10
1.3.3. Öznitelik çıkartımı 14
1.3.4. Sınıflandırma 16
1.3.4 Filtreleme işleme (Medyan filtre) 18
1.3.5 Bağlantılı bileşen etiketleme (Connected component labeling) 20
1.3.6 Histogram eşikleme 21
2. SİSTEMİN GERÇEK ZAMANLI NESNE TAKİBİNDE KULLANILMASI 26
3. DENEYSEL SONUÇLAR 31
3.1. HSV Uzayında Eşikleme Kullanılarak Yapılan Ölçüm 31
3.1.1. Silindirik nesnelerin HSV uzayında eşikleme kullanılarak 0-1m. aralığında uzaklık ölçümü 31
3.1.2. Büyük nesnelerin HSV uzayında eşikleme kullanılarak 1m.-5m. aralığında ölçümü 34
3.1.2.1. Büyük balon referans görüntü 1m. iken ölçüm 34
3.1.2.2. Büyük balon referans görüntü 2,5m. iken ölçüm 37
3.1.2.3. Defter referans görüntü 1,5m. iken ölçüm 38
3.1.2.4. Defter referans görüntü 2,5m. iken ölçüm 42
3.1.3. Küçük nesnelerin HSV uzayında eşikleme kullanılarak 1m.-5m. aralığında ölçümü 43
3.1.3.1. Küçük balon referans görüntü 1m. iken ölçüm 43
3.1.3.2. Küçük balon referans görüntü 2,5m. iken ölçüm 46
3.1.4. 5m., 7,5m., 10m., 12,5m., 15m. için farklı nesneler kullanılarak yapılan ölçüm 47
3.1.4.1. Kırmızı çanta referans görüntü 5m. iken ölçüm 47
3.1.4.2. Kırmızı çanta referans görüntü 10m. iken ölçüm 51
3.2. Sınıflandırma Yöntemi Kullanılarak Yapılan Ölçümler 52

3.3. Histogram Eşikleme Yöntemi Kullanılarak Yapılan Ölçüm 56
3.3.1. Top için histogram eşikleme kullanılarak yapılan mesafe ölçüm sonuçları 56
3.3.2. Silgi için histogram eşikleme kullanılarak yapılan mesafe ölçüm sonuçları 60
3.3.3. Gerçek zamanlı mesafe uyarım çalışması 63
3.4. Başarım Oranları 66
4. SONUÇLAR VE ÖNERİLER 69
KAYNAKLAR 70
EKLER 72
Ek-A 73
KİŞİSEL YAYIN VE ESERLER 77
ÖZGEÇMİŞ 78

ŞEKİLLER DİZİNİ

Şekil 1.1. Rengin beyinde algılanması 6
Şekil 1.2. RGB renk uzayı 7
Şekil 1.3. Toplamsal ana renk kırmızı, yeşil ve mavi 8
Şekil 1.4. Örnek bir resmin C, M, Y ve K kanalları 8
Şekil 1.5. K kanalı çıkarılmış eklenmiş örnek bir resim 9
Şekil 1.6. Konik ve silindirik biçimli HSV renk uzayı 9
Şekil 1.7. Kırmızı balon tespitinde kullanılan görüntü 10
Şekil 1.8. Sırasıyla RGB ve HSV uzayındaki görüntüler 13
Şekil 1.9. Data curser ikonu 13
Şekil 1.10. Farklı piksellerde HSV değerlerinin karşılaştırılması 14
Şekil 1.11. Madeni paralar 20
Şekil 1.12. Eşikleme uygulanmış imge ve tüm paraların farklı renge boyanması 20
Şekil 1.13. İstenen iki para tüm imgeden çıkartılmış görüntü 21
Şekil 1.14. Görüntü ve histogram grafiği 21
Şekil 1.15. Nesnenin ait olduğu piksel kümesi 23
Şekil 1.16. 8 gri seviyeli histogram 23
Şekil 1.17. Üç farklı uzaklık için histogram karşılaştırması 24
Şekil 2.1. Uzaklığın belirlenmesi için geliştirilen yöntemin akış diyagramı 26
Şekil 2.2. Referans görüntü 28
Şekil 2.3. Eşikleme sonucunda elde edilen görüntü 29
Şekil 2.4. Gerçek zamanlı işlem sırasında alınan kamera görüntüsü 29
Şekil 2.5. Gerçek zamanlı görüntünün siyah-beyaz formu 30
Şekil 3.1. Silindirik yüzey gösterildiğinde ölçülen uzaklık: 195,6cm. 32
Şekil 3.2. Taban yüzey gösterildiğinde ölçülen uzaklık: 2193,4cm. 32
Şekil 3.3. Gerçek zamanlı olarak nesnenin kameraya olan uzaklık ölçümü 33
Şekil 3.4. Hata oranları ve ölçümler 34
Şekil 3.5. Referans görüntü 35
Şekil 3.6. 2m.'deki balon görüntüsü 35
Şekil 3.7. 3m.'deki balon görüntüsü 35
Şekil 3.8. 4m.'deki balon görüntüsü 35
Şekil 3.9. 5m.'deki balon görüntüsü 36
Şekil 3.10. Referans 1m. iken alınan sonuçlar 36
Şekil 3.11. Referans görüntü 37
Şekil 3.12. Referans görüntü 2,5m. iken ölçüm sonuçları 38
Şekil 3.13. Referans görüntü 39
Şekil 3.14. İlk eşikleme ve çeşitli işlemler sonucunda elde edilen görüntü 39
Şekil 3.15. 2m.'deki görüntü ve eşikleme sonucu elde edilen görüntü 40
Şekil 3.16. Morfolojik işlemler sonucunda elde edilmiş görüntü 40
Şekil 3.17. Deneysel sonuç grafiği 41
Şekil 3.18. Referans 2,5m. iken deneysel sonuç grafiği 42
Şekil 3.19. Küçük nesne için 1m.' deki referans görüntü 43
Şekil 3.20. Küçük nesne için 2m.'deki görüntü 43
Şekil 3.21. Küçük nesne için 3m.'deki görüntü 43

Şekil 3.22. Connected Component Labeling yöntemi sonucunda elde edilen görüntü 44
Şekil 3.23. Küçük nesne için 4m.'deki görüntü 44
Şekil 3.24. Küçük nesne için 5m.'deki görüntü 45
Şekil 3.25. Referans görüntü 1m. iken ölçüm sonuçları 45
Şekil 3.26. Küçük nesne için 2,5m.'deki görüntü 46
Şekil 3.27. Referans görüntü 2,5m. iken ölçüm sonuçları 47
Şekil 3.28. Referans görüntü 5m.'de çanta 48
Şekil 3.29. Çanta için 7,5m.'deki görüntü 48
Şekil 3.30. Çanta için 10m.'deki görüntü 48
Şekil 3.31. Çanta için 12,5m.'deki görüntü 49
Şekil 3.32. Çanta için 15m.'deki görüntü 49
Şekil 3.33. Referans görüntü 5m. iken ölçüm sonuçları 50
Şekil 3.34. Referans görüntü 10m. iken ölçüm sonuçları 52
Şekil 3.35. 20cm.-110cm. arasında alınan nesnenin referans görüntüleri 53
Şekil 3.36. Oluşturulan sınıflar 55
Şekil 3.37. Referans görüntü ve histogram grafiği 59
Şekil 3.38. 100cm.'deki top için referans görüntü 59
Şekil 3.39. Top için gerçek zamanlı mesafe ölçüm sonuçları 60
Şekil 3.40. Silgi için referans görüntü ve histogram eğrisi 62
Şekil 3.41. 100cm.'deki silgi için referans görüntü 62
Şekil 3.42. Silgi için gerçek zamanlı mesafe ölçüm sonuçları 63
Şekil 3.43. Gerçek zamanlı nesne uyarı bilgisi–'çok uzak' 64
Şekil 3.44. Gerçek zamanlı nesne uyarı bilgisi–'uzak' 65
Şekil 3.45. Gerçek zamanlı nesne uyarı bilgisi–'çok yakın!' 65
Şekil 3.46. 50cm. referans görüntüye göre ölçüm sonuçları 66
Şekil 3.47. 1m. referans görüntüye göre ölçüm sonuçları 66
Şekil 3.48. Piksel/cm grafiği 67
Şekil 3.49. Geliştirilen yönteminin sonuçları 67
Şekil 3.50. Kullanılan yöntemlerin karşılaştırılması 68

TABLOLAR DİZİNİ

Tablo 3.1. Deneyler uzaklık ölçümleri ve oluşan hatalar 33
Tablo 3.2. Referans görüntü 1m. iken nesne uzaklık ölçümü 36
Tablo 3.3. Referans görüntü 2,5m. iken nesne uzaklık ölçümü 37
Tablo 3.4. Referans 1,5m. iken hesaplanan mesafeler ve hata değerleri 41
Tablo 3.5. Referans 2,5m. iken hesaplanan mesafeler ve hata değerleri 42
Tablo 3.6. Referans görüntü 1m. iken nesne uzaklık ölçümü 45
Tablo 3.7. Referans görüntü 2,5m. iken nesne uzaklık ölçümü 46
Tablo 3.8. Referans görüntü 5m. iken nesne uzaklık ölçümü 49
Tablo 3.9. Referans görüntü 10m. iken nesne uzaklık ölçümü 51
Tablo 3.10. 20cm.-110cm. arası her 10cm.'de alınan görüntülerin sisteme tanıtılması 53

SİMGELER DİZİNİ VE KISALTMALAR

B	: Görüntü ortalama değeri
cm.	: Santimetre
D	: Uzaklık
I	: Alınan görüntü
k	: Gözlem Uzayı Boyutu
l	: Öznitelik Uzayı Boyutu
M	: Değişken sayısı
m	: Satır sayısı
m.	: Metre
n	: Sütun sayısı

Kısaltmalar

B	: Blue (Mavi)
C	: Cyan (Cam göbeği)
CMY	: Cyan, Magenta, Yellow (Cam göbeği, Eflatun, Sarı)
G	: Green (Yeşil)
H	: Hue (Renk özü)
HSV	: Hue, Saturation, Value (Renk özü, Doygunluk, Parlaklık)
HSI	: Hue, Saturation, Intensity (Renk özü, Doygunluk, Yoğunluk)
M	: Magenta (Eflatun)
R	: Red (Kırmızı)
RGB	: Red, Green, Blue (Kırmızı, Yeşil, Mavi)
S	: Saturation (Doygunluk)
V	: Value (Parlaklık)
Y	: Yellow

TEK KAMERA KULLANARAK NESNE-KAMERA ARASINDAKİ UZAKLIĞIN BELİRLENMESİ İÇİN BİR YÖNTEM

ÖZET

Bu çalışmada, son yıllardaki teknolojik gelişimlere bağlı olarak, birçok alanda yaygın bir şekilde kullanılan görüntü işleme metotları ile nesne tespiti ve bu nesnenin kameraya olan uzaklığının belirlenmesi üzerinde çalışılmıştır. Tek web kamera kullanarak, daha düşük maliyetle, nesnelerin kameraya olan uzaklıklarını ölçmek için robotik uygulamalarda kullanılabilecek bir yöntem önerilmiştir. Elde edilen sonuçlar üzerinden algoritma optimize edilerek, nesneler arasındaki uzaklığı belirlemede de kullanılabilecek bir algoritma geliştirilmiştir. Sistem gerçek zamanlı olarak test edilmiş ve sonuçlar gözlemlenmiştir. Geliştirilen algoritmanın kullanılan diğer yöntemlerle karşılaştırılması yapılmış ve başarımının daha yüksek olduğu belirlenmiştir.

Anahtar Kelimeler: Görüntü İşleme, Nesne Uzaklık, Tek Kamera ile Uzaklık, Uzaklık Belirleme.

A METHOD FOR DETERMINATION OF OBJECT-CAMERA DISTANCE BY USING SINGLE CAMERA

ABSTRACT

In this study, object detection and determination of object-camera distance are studied by using image processing methods, which are used widely in many areas in recent years depend on technological development. A low cost method that can be used in robotic applications is proposed to measure distance of the object to the camera by using single camera. The results are optimized through a developed algorithm that can also be used to determine the distance between the objects. The system was tested in real-time and results were observed. A comparison with another method was realized and the proposed method was found to be more successful.

Keywords: Image Processing, Object-Distance, Distance with Single Camera, Determination Distance.

GİRİŞ

Günümüzde insanların sürekli yer değiştirmesi ve hareket halinde olmasından dolayı, sabit görevleri yerine getiren robotların beklenildiği kadar faydalı olmadığı görülmüş ve bu yüzden daha kullanışlı mobil robotların tasarlanması ve üretilmesi amaçlanmıştır. Mobil robotların çevrelerini tanıması, bulundukları ortamdan bilgi toplaması ve bu bilgiler doğrultusunda karar verme mekanizmalarının geliştirilmesi gerekmiştir. Robotların bu gereksinimlerini karşılayabilmeleri için, çevrelerini algılayabilecek farklı özellikteki sensörlere ihtiyaç duyulmuştur [1]. İnsanın, çevresini keşfetmesi ve hareket edebilmesi için ihtiyaç duyduğu görme duyusunun kamera kullanımı sayesinde mobil robotlara aksettirilmesiyle, görme tabanlı mobil robot çalışmalarının önü açılmıştır. İnsan gözü ile nesnelerin fark edilmesine benzer şekilde, nesnelerin bize olan uzaklıklarının, büyüklüklerinin ve hareketlerinin, kamera kullanımı yoluyla sağlanmasına yönelik çalışmalar son yıllarda ağırlık kazanmıştır. Bu amaca yönelik olarak, gerekli olan görüntünün işlenmesi çok vakit aldığından, çoğu robot sisteminde, kamera haricinde görsel olmayan sensörler de robot kontrolleri için kullanılmaktadır. Ancak, nesnelerin robota olan uzaklıklarının tespiti için, bu sensörlerden çok sayıda kullanılmasına duyulan ihtiyaç, maliyetin artması açısından bir dezavantaj olarak ortaya çıkmaktadır. Dolayısıyla, pratikte, robot kontrollerinde kamera kullanılması daha avantajlı bir duruma gelmiştir. Kamera kullanımıyla elde edilen diğer bir üstünlük ise sadece nesnelerin bize olan uzaklıklarının tespiti değil, aynı zamanda nesnelerin renkleri, desenleri ve şekilleri hakkındaki bilgilerin de beraberinde elde edilebilmesidir [2].

Bu tez çalışması kapsamında, görüntü işleme yöntemleri kullanılarak nesne tespiti ve ilgili nesnenin kameraya olan uzaklığının belirlenmesi üzerinde çalışılmıştır. Çeşitli nesne büyüklükleri için algoritmalar geliştirilmiştir. Elde edilen sonuçlar üzerinden algoritma optimize edilerek, nesne uzaklık belirlemede kullanılabilecek bir algoritma geliştirilmiştir. Sistem gerçek zamanlı olarak test edilmiş ve sonuçlar gözlemlenmiştir. Geliştirilen algoritma ile nesnenin kameraya olan uzaklığı %3-4 hata oranı ile belirlenebilmektedir.

Bölüm 1'de, görüntü işleme yöntemleri, renkli nesnelerin belirlenmesi, tez çalışmasında kullanılan görüntü işleme yöntemleri hakkında teorik bilgi ve nesne-uzaklık belirleme algoritması tek imge üzerinden detaylı bir şekilde anlatılmaktadır. Bu tez çalışması kapsamında kullanılan temel görüntü işleme yöntemleri incelenmiştir.

Bölüm 2'de, gerçek zamanlı olarak nesne uzaklığını belirlemenin nasıl yapıldığı anlatılmaktadır. Yine tek görüntü için matematiksel uzaklık hesabı verilmektedir.

Bölüm 3'te elde edilen deneysel sonuçlar incelenmiş olup, geliştirilen yöntem farklı büyüklük ve şekildeki nesnelere uygulanarak sonuçlar karşılaştırılmıştır. Hatalı sonuçlar üzerinden iyileştirilen algoritma geliştirilmiştir.

Bölüm 4'te, yapılan çalışma sonucu elde edilen başarım belirtilmiştir. Önerilere yer verilmiştir.

Bu çalışmanın amacı, tek kamera kullanılarak düşük maliyetli nesne uzaklık ölçümü için görüntü işleme yöntemlerinin robotik çalışmalarda gerçek zamanlı uygulanabilirliğini araştırmaktır. Çalışmada renkli nesnelerin kameraya olan uzaklığını belirlemede kullanılabilecek bir yöntem geliştirilmiştir.

1. GÖRÜNTÜ İŞLEME VE NESNE TESPİTİNDE KULLANILAN YÖNTEMLER

Görüntü işleme, ölçülmüş veya kaydedilmiş olan dijital görüntü verilerini, bilgisayar ve yazılımlar yardımıyla istenen şekilde değiştirmeye yönelik yapılan bilgisayar çalışmasıdır. Teknolojik gelişmelerle birlikte, günümüzde görüntü işleme ve bilgisayarlı görü uygulamaları da hızla gelişerek robotik, fabrika otomasyonu, tıbbi çalışmalar, askeri uygulamalar, akıllı arabirimler, tarım-hayvancılık, trafik, uzaktan algılama uygulamaları ve güvenlik sistemleri gibi birçok alanda yerini almıştır. Kalite kontrol, sınıflandırma, ayırma gibi işlemlerin aktif olarak yapıldığı uygulamaların çoğunda görüntü işleme yöntemlerinin yaygın olarak kullanıldığı görülmektedir. Bilgisayar kullanımının insan hayatının hemen hemen her noktasına yoğun bir şekilde girmesiyle, günlük yaşantımızda karşılaştığımız problemlerin algılanması ve çözümlenmesi zorunluluğu ortaya çıkmıştır. Bu bağlamda, görüntü işlemenin hareket analizi, yer değiştirme, hız, nesne takibi ve nesne tanımlanması gibi farklı alanlarda etkili bir şekilde kullanılması yaygın araştırılan bir konu haline gelmiştir.

Görüntü işleme uygulamaları çok yaygın bir şekilde birçok alanda kullanılmaktadır. Yıllardan beri görüntü işleme çalışmalarında, farklı uygulamaların sayısının artmaya devam ettiği bilinmektedir. Yapılan literatür çalışması sonucunda, nesne bilgileri hakkında da değişik görüntü işleme uygulamalarının kullanıldığı görülmüştür. Hareket tespitine ilgi 1970'lerin sonlarından itibaren ivme kazanmıştır. 1979'da Philadelphia'da yapılan ilk özel çalışma konferansı bu konuya öncü olmuştur. Bu adımdan sonra bir çok konferans, sempozyum ve çalıştay yapılmıştır. NATO Advanced Study Instute 'Image Sequence Processing and Dynamic Scene Analysis' isimli özel konferans 1982'de yapılmıştır. Aynı sene ilk uluslararası konferans 'Time-Varying Processing and Moving Object Recognation' ismi ile Florence, İtalya'da yapılmıştır [3].

İnsanların güvenlik istekleri gün geçtikçe artmaktadır. İnsan-bilgisayar etkileşiminin artması ve bilgisayarlı otomasyon sistemlerinin insan hayatına etkisi düşünüldükçe,

sağlıklı bir güvenlik sisteminin temel taşlarından birisi haline gelen gözetleme sistemleri daha da önem kazanmaktadır. Bir güvenlik sistemi için en temel bileşen, genellikle bir gözcü veya gözetleme sistemi olarak ifade edilebilir. Böyle bir sistem, elektronik olarak veya bilgisayar-temelli bir yapı ile, insanın bakmadığı zamanlarda da görüntü işleme tekniklerini kullanarak gözetlenmesi istenilen bölgeyi gözetlemekte ve hareketli nesnelerin tespit ve takibini (T-ve-T) yapabilmektedir. İnsan görüsü ile bilgisayarlı görü arasında temel bazı fonksiyonel benzerlikler bulunmaktadır. Bu iki sistemin de temel amacı uzamsal (konuma bağlı) veriyi yorumlayabilmektir. Böyle bir veri birden çok boyut ile endekslenebilmekte ve büyük boyutlu depolama isteklerine sahip olmaktadır.

Hareketli nesne veya nesneleri tespit ve sonrasında da takip etme, bilgisayarlı görü kapsamında ele alınan büyük bir problemdir. Bilgisayarlı görü problemlerinin yorumlanmasında bir kesinlik bulunmamaktadır. Bu sebeple, videolarda hareketli nesne T-ve-T sürecinin belirli aşamalarının alt problemlere ayrıştırılması daha etkin bir yolla çözüm üretmeyi sağlamaktadır. Bu tarz alt problemler bir arada değerlendirilerek ana problemin çözümü gerçekleştirilebilmektedir. Renk tabanlı nesne kestirimi için RGB (Red, Green, Blue) renk uzayındaki görüntü farklı renk uzaylarına çekilerek yapılmaktadır. Bu işlemler genellikle HSV (Hue, Saturation, Value) renk uzayında yapılmaktadır [4].

1.1. Renk Tabanlı Yaklaşımlar

Sayısal bir resim (görüntü), nesneler tarafından yansıtılan ışık enerjisinin (analog sinyal) bir algılayıcı tarafından öngörülen elektromanyetik aralıkta algılanarak sayısal sinyal haline dönüştürülmesi ile oluşmaktadır. Bir görüntünün temel bileşeni pikseldir. Sayısal bir görüntü *mxn* boyutlu piksellerden oluşan bir matris ile ifade edilir. Gri tonlu görüntülerde, görüntü farklı gri ton değerlerinden oluşur. Gri değer aralıkları $\{0,1,2,3,...255\}$ şeklinde ifade edilir. Bunun anlamı, bir gri tonlu görüntüde 256 tane farklı gri ton değeri, daha doğrusu gri değer bulunmaktadır. Sıfır gri değeri kural olarak siyah renge, 255 gri değeri ise beyaza karşılık gelmektedir. Bu değerler arasında ise gri tonlar oluşmaktadır.

Elektromanyetik spektrumda 0,4-0,5 μm dalga boyu mavi renge; 0,5-0,6 μm dalga boyu yeşil renge; 0,6-0,7 μm dalga boyu kırmızı renge karşılık gelmektedir. Bu dalga

boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası ile kırmızı-yeşil-mavi (RGB) kombinasyonunda üst üste düşürülerek renkli görüntü elde edilmektedir.

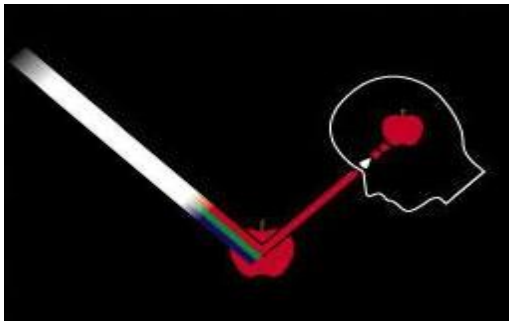
Video kamera ve tarayıcı gibi görüntü yakalayıcı cihazlarla alınarak sayısallaştırılan görüntülerin, geliştirilen yöntemlerle işlenip analiz edilmesi, görüntülerden bilgi elde edilmesi ve bu bilgilerin yorumlanması, görüntü işleme alanının temelini oluşturmaktadır.

Görüntü işleme uygulamalarına; fabrika otomasyon uygulamalarından endüstriyel kalite/üretim kontrolleri, tıbbi uygulamalardan kan hücrelerinin sayımı, bilgi işlem uygulamalarından karakter tanıma, güvenlik amaçlı uygulamalardan yüz tanıma, askeri uygulamalardan akıllı silahların üretimi gibi örnekler verilebilir [4].

Dolayısıyla gelişen teknoloji ile, önceden çok pahalıya mal edilen sistemlerin yaptığı işler, görüntü işleme sistemleri ile daha hızlı, ekonomik ve hassas bir şekilde yerine getirilebilir. Görüntü işleme uygulamalarının amacı aktif dünyayı algılama ve yorumlama yeteneğine sahip sistemleri görmeye dayalı olarak geliştirmektir. Bu sistemlerde kamera ile görüntü dış ortamdan alınır, görüntü işleme teknikleri ile işlenir, yorumlanır ve yapılması gereken işleme karar verilir [5].

1.2. Renk Modelleri

Renk, ışığın değişik dalga boylarının gözün retinasına ulaşması ile ortaya çıkan bir algılamadır. İnsan tarafından renklerin algılanması, ışığın cisimler tarafından yansıtılmasıyla ve nesnenin göz yardımıyla beyne iletilmesi sayesinde gerçekleşir. Bu algılama, ışığın maddeler üzerine çarpması ve kısmen soğurulup kısmen yansıması nedeniyle çeşitlilik gösterir ki bunlar renk tonu veya renk olarak adlandırılır. Renk tamamen öznel ve kişisel bir olgudur. Renk modellerinin amacı, insanlar, makineler veya bilgisayar programları arasında renkleri tarif etme işlemine yardımcı olmaktır. Renk, beyindeki özel görsel uyarıcılarla olan tepkisidir. Şekil 1.1'de rengin beyinde algılanması gösterilmiştir.



Şekil 1.1. Rengin beyinde algılanması

Buna rağmen renk, renk tayfı üzerindeki enerji dağılımını ölçerek tanımlanabilir. İnsan gözünün retinası kabaca kırmızı, yeşil ve mavi ışığı temsil eden üç algılayıcı türü ile renkleri algılar. Renklere duyarlı hücrelerden gelen sinyaller, renk yoğunluğuna duyarlı hücrelerden gelen sinyallerle birleşerek beyinde çeşitli algıların oluşmasını sağlar. Renk algısının tarifi sırasında kullanılan bazı terimleri bilmek gerekir. Bunlar;

Parlaklık (Brightness): Görüntüdeki ışığın miktarıdır. Rengin aydınlığını, yani içindeki beyaz oranını belirler. 0 – 100 arasında değişir.

Renk Özü (Hue): Rengin baskın dalga uzunluğunu belirler, örneğin sarı, mavi, yeşil renklere yakınlığı olarak söylenebilir. Açısal bir değerdir. 0° - 360° ile ifade edilirken, bazı uygulamalarda ise 0 - 100 arası normalize edilir.

Renklilik (Colourfulness): Görüntüdeki renklerin renk özü oranlarının miktarıdır.

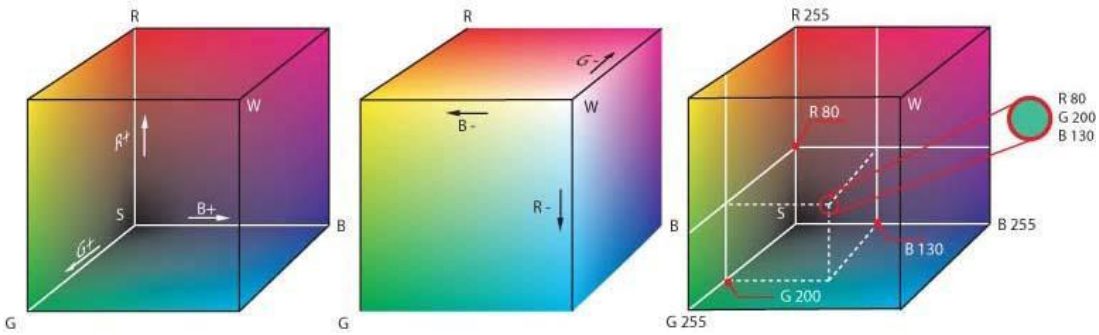
Işıklılık (Lightness): Belirli bir beyaz ışığa göre görüntüdeki parlaklık miktarıdır.

Doygunluk (Saturation): Görüntünün parlaklığına bağlı renkliliğidir. Rengin "canlılığını" belirler. Yüksek doygunluk canlı renklere neden olurken, düşük doygunluk rengin gri tonlarına yaklaşmasına neden olur. 0 - 100 arasında değişir [6].

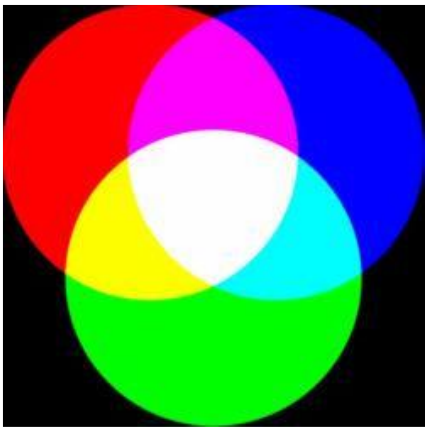
Renk modeli, renkleri tanımlamak, oluşturmak ve görüntülemek için kullanılan bir yöntemdir. Renkler; parlaklık, renk özü ve renklilik özellikleri ile tanımlanabilir. Bilgisayarlar renkleri, rengi oluşturan kırmızı, yeşil ve mavi miktarlarına göre tanımlarlar [7].

1.2.1. RGB renk modeli

RGB renk modeli, İngilizcedeki 'Red' 'Green' 'Blue' ('Kırmızı' 'Yeşil' 'Mavi') kelimelerinin baş harflerinden ismini alan bir renk sistemidir. Şekil 1.2'de RGB renk uzayı gösterilmiştir. En sık kullanılan renk modelidir. Bunun sebebi, 1953'te ilk renkli fotoğraf makinesi Polaroidde ve ondan sonra da televizyonlarda standart olarak kabul edilmiş olmasıdır. Işığı temel alarak, doğadaki tüm renklerin kodları bu üç temel renge referansla belirtilir. Her renk %100 oranında karıştırıldığında beyaz ve %0 oranında karıştırıldığında siyah elde edilir. Bu modelin temeli, insanın göz retinasında bu renklere rastgelen ışık dalga boyu sensörleridir. Bunların arasında kalan dalga boylarında da bu üç sensörün her biri değişik seviyelerde tepki verir ve bu tepki beyinde renk algısını oluşturur. Örneğin gökkuşağının sarı olarak adlandırılan dalga boyunda bir ışık göze düştüğünde, ağırlıklı olarak 'kırmızı' ve 'yeşil' sensörler uyarılır. Beyinde bu birleşim 'sarı' renge dönüşür. Kırmızı ışık geldiğinde ise sadece 'kırmızı' sensörler uyarılır. Hem kırmızı hem yeşil hem de mavi ışığın aynı anda gelmesi ile tüm dalga boylarının aynı anda gelmesi aynı etkiyi oluşturarak beyaz ışık şeklinde algılanmaktadır. Şekil 1.3'te toplamsal ana renk gösterimi verilmiştir. Bu sebeple, şiddetleri sönük ve parlak olarak ayarlanabilen, bu üç renkte ışık kaynaklarının bulunduğu şartlarda tüm renkleri elde etmek mümkündür [7].



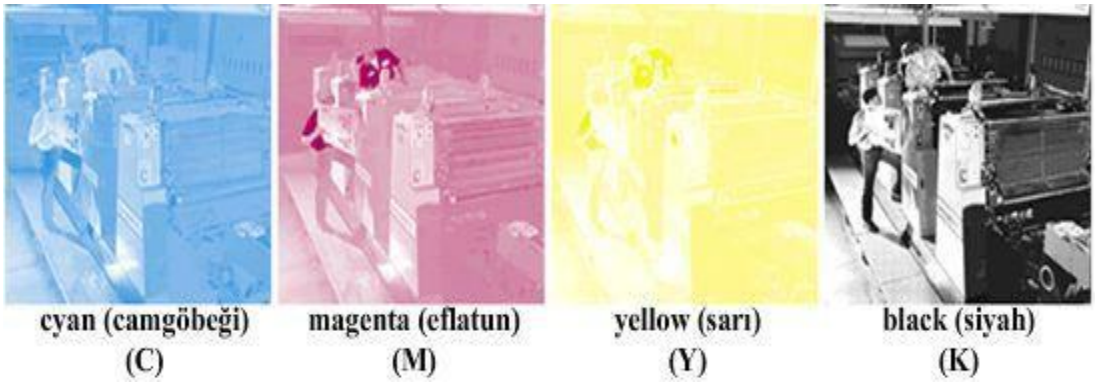
Şekil 1.2. RGB renk uzayı



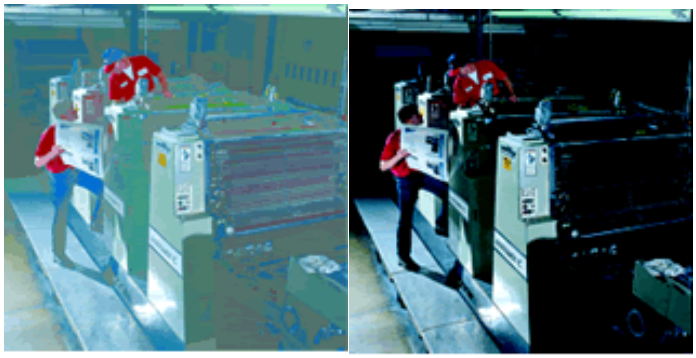
Şekil 1.3. Toplamsal ana renk kırmızı, yeşil ve mavi

1.2.2. CMY/CMYK renk modeli

CMYK, matbaa baskısında dört temel işlem renginin kısaltmasıdır (Cyan (Camgöbeği), Magenta (Eflatun), Yellow (Sarı), Black (Siyah)). Şekil 1.4 ve Şekil 1.5'te CMY/CMYK renk modeli gösterilmiştir. Aslında temel renk sayısı üçtür. Siyah bu renklere zorunlu olarak ilave edilmiştir. Kuramda üç rengin karışımının siyahı oluşturması gerekirken, pratikte bu durum böyle değildir. Hem üç rengin mürekkepleri yeterli renk şiddetini sağlamadıklarından hem de üç renkli mürekkebin karışımı yerine siyah mürekkep kullanmanın maliyetinin daha düşük olması nedeniyle siyah renk sisteme dahil edilmiştir [7].



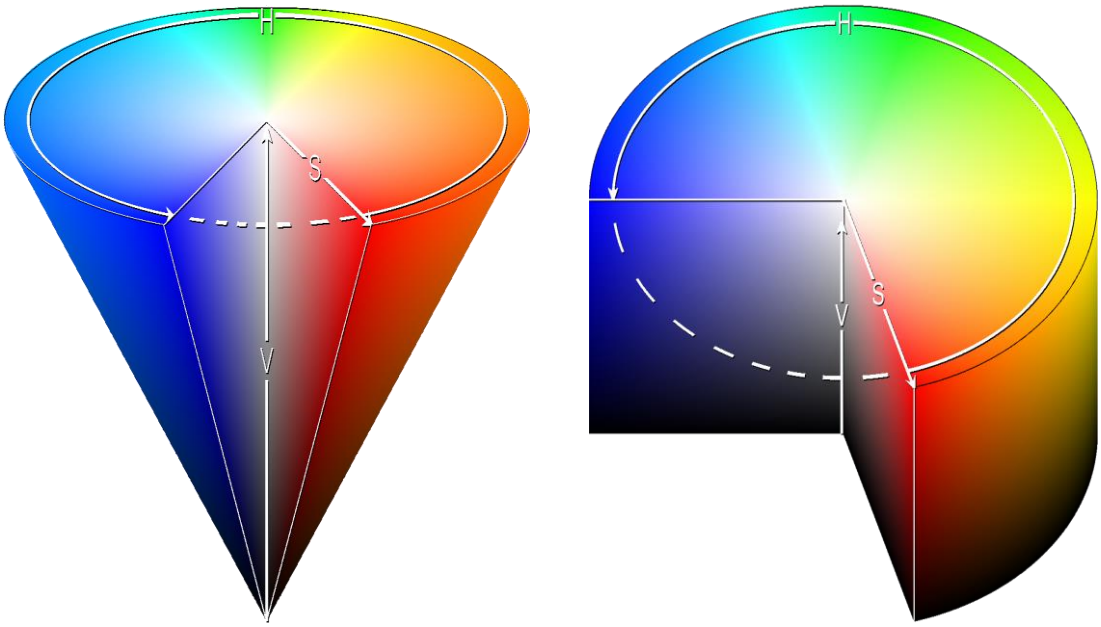
Şekil 1.4. Örnek bir resmin C, M, Y ve K kanalları [7]



Şekil 1.5. K kanalı çıkarılmış eklenmiş örnek bir resim [7]

1.2.3. HSV renk modeli

HSV (Hue, Saturation, Value) veya HSB (Hue, Saturation, Brightness) renk uzayı, renkleri sırasıyla renk özü, doygunluk ve parlaklık olarak tanımlar. HSV renk uzayının kullanılma amacı RGB uzayına göre insan görü düzeneğine daha yakın bir yapı oluşturmaktır. HSV, RGB renk uzayından doğrusal olmayan bir dönüşüm ile elde edilmektedir. HSV, aygıt bağımlıdır. Yani bu uzayda tanımlı bir renk, rengi üreten aygıt cihazına göre değişim gösterebilir. Şekil 1.6'da HSV renk uzayı gösterilmiştir.



Şekil 1.6. Konik ve silindirik biçimli HSV renk uzayı [7]

1.3. Nesne Tespitinde Kullanılan Görüntü İşleme Yöntemleri

Tez çalışmasının ilk aşaması olarak, alınan görüntüde takip edilecek nesne belirlenmektedir. Daha sonra, nesnenin sabit konumlandırılmış kameraya olan uzaklığı hesaplanmaktadır. Nesne olarak kırmızı bir balon kullanılmıştır. Balon farklı mesafelerde görüntülenerek, elde edilen görüntüler üzerinden balonun net olarak belirlenmesi ve daha sonra da kameraya olan uzaklığı üzerinde çalışılmıştır. Nesne tespiti için üzerinde çalışılan kırmızı balon görüntüsü Şekil 1.7'de verilmektedir.



Şekil 1.7. Kırmızı balon tespitinde kullanılan görüntü

1.3.1. Kullanılan görüntü işleme uygulamaları

Görüntü işleme uygulamaları olarak, HSV renk modelinde eşik belirleyerek sınıflandırma yöntemi kullanılması ve histogram değerlerindeki değişimler kullanılarak nesnelerin mesafe bilgisi elde edilmesi uygulamaları gerçekleştirilmiştir.

1.3.2. RGB den HSV ye aktarılma

HSV renk uzayı renkleri içerdiği RGB değerlerine göre değil, Hue (Renk Özü), Saturation (Doygunluk) ve Value (Parlaklık) değerlerine göre belirten bir renk uzayı

türüdür. 1978 yılında Alvy Ray Smith tarafından RGB uzayına göre insan görme sistemine daha benzer bir yapı oluşturmak amacıyla tanımlanmıştır. H değeri 0-360° arasında değişirken, renklerin öz değerleri de değişir. Ancak H sabit olarak seçilir ve diğer değerler (S ve V) 0-100 arasında değiştirilirse aynı rengin farklı doygunluk ve parlaklıktaki değerleri elde edilir. Bu özelliğinden dolayı HSV, renk temelli ayırma işlemlerinde sıklıkla tercih edilir. RGB renk uzayında parlaklığa bütün bileşenlerin etkisi bulunmakla beraber, HSV uzayında ise RGB uzayından farklı olarak, parlaklığa sadece H değerinin etkisi vardır. Bu yüzden V değeri boyunca gidildiğinde renk bileşeni sabit kalmaktadır. Geliştirilen sistemde, kameradan alınan görüntü RGB görüntüdür. Öncelikli olarak kameradan alınan görüntü RGB renk uzayından HSV renk uzayına dönüştürülmüştür. RGB uzayından HSV uzayına dönüştürülmüş resimler üzerinde, deneme yanılma yoluyla belirlenen bir eşik uygulanmıştır. Alınan resimlerden H ve S değeri belirlenen eşiğin üstünde olan piksellerden nesne kestirimi yapılmıştır.

Şekil 1.7'de görülen kırmızı balonun tespiti için, ilk olarak görüntü Matlab ortamına aktarılmıştır. Görüntünün bulunduğu klasör ve üzerinde çalışılan Matlab programı aynı klasör olmalıdır. Aksi takdirde görüntü Matlab ortamına aktarılmayınca program hata verecektir. Matlab ortamına aktarılan ilk görüntü RGB formatındadır. Kırmızı balonun daha kolay belirlenmesi ve ortam ışık şiddeti değişikliklerinin geliştirilen algoritmaya daha az etki etmesi için RGB uzayındaki görüntü HSV uzayına geçirilmiştir. Görüntü işleme uygulamalarında genellikle HSV uzayında çalışılmaktadır.

RGB renk uzayından HSV renk uzayına aktarılma işleminin matematiksel olarak gösterimi aşağıdaki gibidir. R, G, B değerlerinin aralığı 0-255'ten 0-1'e değiştirmek için 255'e bölünerek elde edilir [8]. Denklem (1.1)-Denklem (1.9)'a kadar matematiksel denklemleri gösterilmektedir.

$$R' = R/255 \tag{1.1}$$

$$G' = G/255 \tag{1.2}$$

$$B' = B/255 \tag{1.3}$$

$$Cmax=max(R', G', B') \tag{1.4}$$

$$Cmin=min(R', G', B') \tag{1.5}$$

$$\Delta=Cmax-Cmin \tag{1.6}$$

Hue hesaplama;

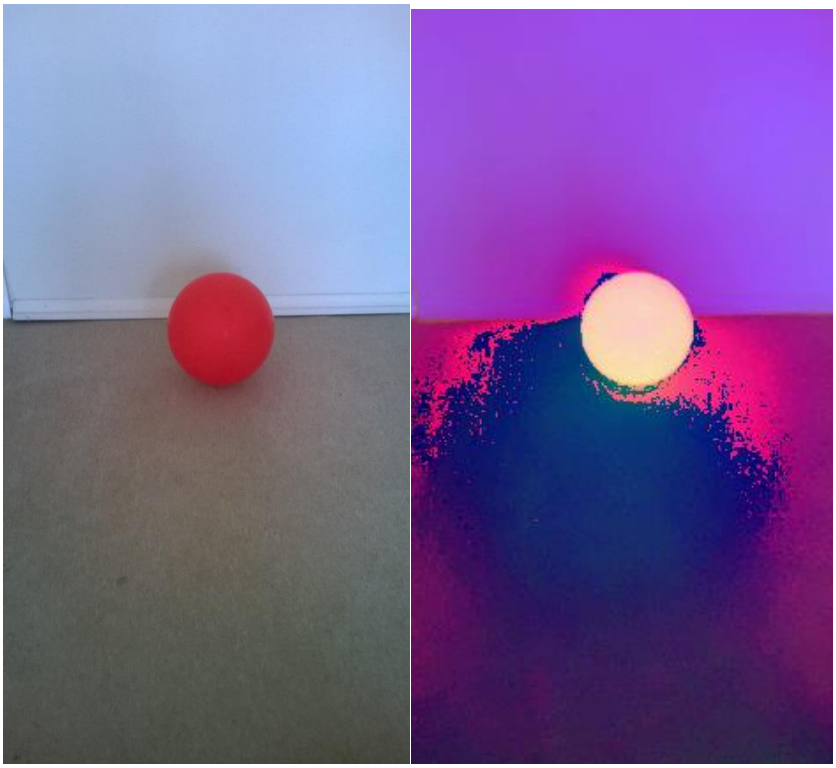
$$H=\begin{cases} 0^{\circ} & \Delta=0 \\ 60\times\left(\frac{G'-B'}{\Delta}\bmod 6\right) & Cmax=R' \\ 60\times\left(\frac{B'-R'}{\Delta}+2\right) & Cmax=G' \\ 60\times\left(\frac{R'-G'}{\Delta}+4\right) & Cmax=B' \end{cases} \tag{1.7}$$

Saturation hesaplama;

$$S=\begin{cases} 0 & Cmax=0 \\ \frac{\Delta}{Cmax} & Cmax^{1}0 \end{cases} \tag{1.8}$$

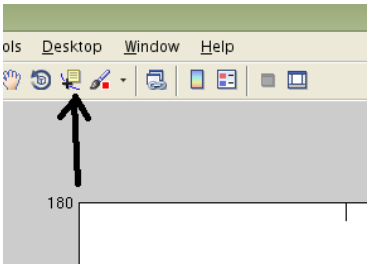
Value hesaplama;

$$V=Cmax \tag{1.9}$$



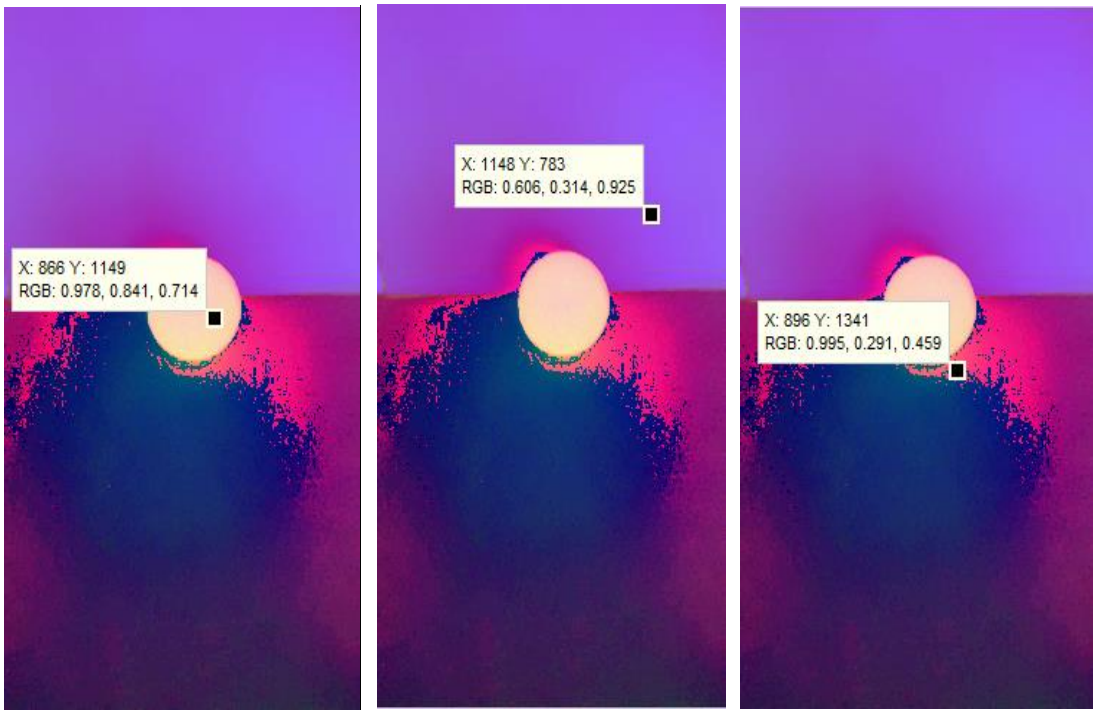
Şekil 1.8. Sırasıyla RGB ve HSV uzayındaki görüntüler

Şekil 1.8'de gösterilen HSV uzayındaki görüntüden piksel değerleri incelenerek uygun bir eşik belirlenerek kırmızı balon tespiti yapılmaktadır. Görüntüde hangi pikselde hangi renk değerleri ve parlaklık değerleri olduğu Şekil 1.9'da gösterilen "data curser" ikonu ile yapılmaktadır. Bu ikon görüntü işleme, işaret işleme gibi Matlab'da yapılan çoğu uygulamada verilerin incelenmesinde oldukça sık kullanılmaktadır.



Şekil 1.9. Data curser ikonu

Data curser ikonu kullanılarak HSV uzayındaki piksel değerleri incelenerek nesne tespiti için uygun eşik belirlenmiştir.



Şekil 1.10. Farklı piksellerde HSV değerlerinin karşılaştırılması

Şekil 1.10'da görüldüğü üzere farklı piksellerde görüntünün HSV değerleri farklılık göstermektedir. Kırmızı balonun tespitinin yapılabilmesi için balon bölgesi pikselleri ile diğer bölgelerdeki pikseller karşılaştırılmıştır. Bu çalışmada ikinci renk kanalı olan S (saturation) eşik belirlemek için kullanılmıştır. Şekil 1.10'da görüldüğü gibi kırmızı balonun olduğu bölgelerde saturation kanalı (S) renk bilgileriyle diğer bölgelerdeki renk bilgileri farklılık göstermektedir. Farklılık göstermesi kırmızı balon tespitini kolaylaştırmaktadır. Balonun tespiti için HSV görüntüde 2. Kanal için eşik aralığı 0,7-0,95 arası olarak belirlenmiştir.

1.3.3. Öznitelik çıkartımı

Öznitelik, temel olarak örüntüye ait ölçülebilir ya da gözlenebilir bilgi şeklinde ifade edilebilir. Öznitelik çıkarımı, ilgisiz ve fazla bilgiyi eleyerek örüntüye ait karakteristik özelliklerin elde edilmesini sağlar. Temel anlamda bir çeşit boyut indirgeme işlemidir. Özellikle görüntü işleme uygulamalarında, çok yüksek boyutlu veriler kullanıldığından, bu verilerin orijinal halleriyle kullanılması işlem süresini büyük oranda artırmaktadır. Öznitelik çıkarım yöntemleriyle, orijinal verilerin gereksiz ve yüksek boyuttaki bilgileri elenip karakteristik özellikleri korunarak, daha küçük boyutlara indirgenmesi sağlanır [9]. Örüntü tanıma aşamalarından biri olan öznitelik

çıkarımı, sistemin başarısında önemli bir rol oynamaktadır. Uygun seçilmiş öznitelikler sınıfların karakteristik özelliklerini ifade ederler ve tanıma başarısını olumlu yönde etkilerler. Öznitelik çıkarımının amacı, sınıfa ait bilginin mümkün olan en fazlasını daha küçük boyutlara taşıyabilmektir. Başka bir ifadeyle öznitelik çıkarımının amacı k-boyutlu bir X gözlem uzayından, daha küçük l-boyutlu Y öznitelik uzayına, X uzayını mümkün olan en iyi şekilde ifade edebilen bir T dönüşümü bulabilmektir [10]. Bu dönüşüm Denklem (1.10)'da gösterilmektedir.

$$X[k \times k] \xrightarrow{T} Y[l \times l] \qquad k>l \tag{1.10}$$

Görüntü işleme uygulamalarında öznitelik çıkarımı, sayısal görüntülerin şekillerinin belirlenmesi, tanınması ve sınıflandırılması gibi uygulamalarda kullanılmaktadır.

Öznitelik çıkarma işleminde, örüntü hakkındaki ilgisiz bilgilerin elenip karakteristik özelliğin elde edilmesiyle belli oranda boyut indirgeme sağlanır [11]. Ancak, boyut indirgeme sonucu elde edilen özniteliklerin tamamı gerekli olmayıp ifade sınıflandırması için önemi ve yararı olmayan bilgiler içermektedirler. Dolayısıyla önemli özniteliklerin seçilmesi kullanılan sınıflandırma algoritmaların hızlı çalışmasını, zaman kazancını ve belleğin gereksiz yere kullanılmamasını sağlayacaktır [12]. Öznitelik seçiminde çıkarılmış olan özniteliklerin ayırt edicilikleri çeşitli yöntemlerle incelenerek mevcut öznitelik kümesinden daha ayırt edici bir altküme bulunması amaçlanmaktadır [13].

Nesne sınıflandırmada öznitelik çıkarımı kritik bir rol oynamaktadır. Çıkarılan özniteliğin sınıflandırılacak olan nesneyi ayırt edici kılması gerekmektedir. Görsel özniteliklerin kullanılmasındaki en önemli sebep, her bir nesnenin kendine özel görsel özniteliklerinin olmasıdır. Öznitelik çıkarımı, nesne temsili ile yakından ilgilidir. Örneğin histogram tabanlı temsillerde nesnenin renk özniteliği kullanılırken, çevre çizgisi tabanlı temsillerde nesnenin kenar özniteliği tercih edilmektedir. Nesne sınıflandırmada genellikle renk, şekil, kenar, optik akış ve doku gibi öznitelikler tercih edilmektedir. Sınıflandırmada kullanılan özellikler, aşağıda ayrıntılı olarak açıklanmıştır.

Renk: Bir nesnenin rengi, ışık kaynağının ışık tayfının güç dağılımı ve nesnenin yüzey yansıma özellikleri olmak üzere iki esas fiziksel faktörden etkilenmektedir. Görüntü

işleme çalışmalarında, rengi temsil etme amacıyla genellikle RGB renk uzayı kullanılmaktadır. Bunun yanında YUV ve HSV uzayları da kullanılmaktadır. Nesne sınıflandırmada hangi renk uzayının kullanılmasının daha verimli olduğuna dair kesin bir karar bulunmamaktadır.

Kenar: Nesne dış çizgileri görüntü yoğunluğunda genellikle büyük değişimlere yol açmaktadır. Kenarların en önemli özelliği renk öznitelikleri ile kıyaslandığında ışık değişimlerine daha az duyarlı olmasıdır. Nesnelerin dış çizgisini kullanan algoritmalar öznitelik olarak genellikle kenar özniteliğini kullanmaktadır. Sonuçlarının basitliği ve kesinliği sebebiyle en çok tercih edilen kenar bulma yaklaşımı Canny Kenar Bulma (Canny Edge Detector) yöntemidir.

Optik Akış: Optik akış, bir bölgedeki her bir pikselin ötelemesini tanımlayan yer değiştirme vektörlerinin yoğunluk alanıdır. Arka arkaya gelen görüntüdeki eş piksellerin parlaklıkları kullanılarak hesaplanmaktadır.

Doku: Doku, pürüzsüzlük ve düzenlilik gibi öznitelikleri ölçen yüzeyin yoğunluk değişim ölçüsüdür. Renk ile karşılaştırıldığında doku tanımlayıcılarını oluşturabilmek için filtre uygulama gibi ön işlemlere ihtiyaç duymaktadır. Kenar öznitelikleri gibi doku öznitelikleri de ışık değişimlerine daha az duyarlıdır [14].

Yapılan çalışma kapsamında, görüntü işleme kullanılarak sınıflandırma yapılabilmesi için, görüntülerden öznitelik çıkartılması gerekmektedir. RGB renk uzayında alınan görüntünün R, G, B ortalamaları birer öznitelik olarak seçilmiştir. Bu sayede 3 boyutlu bir özellik uzayı elde edilmiştir. RGB uzaylarının öznitelik belirlemede yetersiz olduğu noktalarda görüntüler HSV uzayına çekilerek öznitelik çıkartmada H,S,V uzayları kullanılmıştır.

1.3.4. Sınıflandırma

Sınıflandırma, bir veri grubu içinde belirli bir sınıfı oluşturan objelerin benzerliğinden yola çıkarak ve özelliklerine göre seçilerek gruplandırılması olarak tanımlanabilir. Sınıflandırma, birçok bilim dalında kullanılan bir karar verme işlemidir. Görüntü sınıflandırma işleminde amaç, bir görüntüdeki bütün pikselleri çalışma uzayında karşılık geldikleri sınıflar veya temalar içine otomatik olarak atamak, yerleştirmektir.

Diğer bir anlamda, görüntüdeki objelerin segmentasyonudur. Sınıflandırma ile görüntüdeki objeler, belirlenen ya da denetimsiz sınıflandırmada olduğu gibi sınıflandırıcı tarafından belirlenen sınıflara ayrılarak tanımlanırlar.

Sınıflandırma işleminde dikkat edilecek konular şu şekilde sıralanabilir:

- Algılayıcı, algılama zamanı ve spektral bantların amaca uygun olarak seçimi,
- Çalışma alanlarının uygun seçilmesi,
- Amaca yönelik sınıflandırma algoritmalarının seçimi,
- Belirlenen bu özelliklerin tüm görüntüye uygulanması ve sonuç görüntülerinde doğruluk analizlerinin yapılması.

Sınıflandırmaların sonucunda ise, iki tip hataya rastlanmaktadır. Bunlar;

- Piksellerin yer alması gerekenden farklı bir sınıfa atanması (error of comission),
- Piksellerin uygun bir sınıfa atanamamasıdır (error of omission).

Bu sebeplerden dolayı, sınıflandırmayı yapmadan önce, amaca en uygun ve yukarıda bahsedilen iki hatayı en aza indirecek olan sınıflandırma yönteminin belirlenmesi gerekmektedir.

İki tip sınıflandırma yöntemi vardır:

a. Denetimli Sınıflandırma (Supervised Classification)
b. Denetimsiz Sınıflandırma (Unsupervised Classification)

Denetimsiz sınıflandırma yöntemi piksellerin, kullanıcı müdahalesi olmadan algoritmalar yardımı ile otomatik olarak kümelendirilmesi temeline dayanmaktadır. Denetimsiz sınıflandırma yöntemleri içerisinde en yaygın olarak kullanılan ISODATA (Iterative Self Organizing Data Analyses Technique, Tekrarlı veri Analizi Yöntemi) dır [15].

Bu yöntem karar kuralı olarak, minimum uzaklığı kullanır. Pikseller, görüntünün sol üst köşesinden başlanarak soldan sağa ve satır satır analiz edilir. Aday piksel ile her bir küme ortalaması arasında spektral uzaklık hesaplanır ve en yakın kümeye atanır. Öncelikle istenilen sınıf sayısı kadar oluşturulan kümenin ortalaması hesaplanır. Denetimsiz sınıflandırma; görüntüdeki veri tanımlanamadığında başvurulan yöntemdir. Bu yöntemde analizi yapan kişinin sınıflandırılmış görüntüyü, verinin

hangi kümeye ait olduğunu bulabilmek için başka bir referans bilgisi ile kıyaslama yapması gerekmektedir.

Bu uygulamada nesnelerin RGB görüntülerinden HSV görüntülerine geçilmiştir. HSV uzayında nesnelere ait öznitelikleri çıkartılarak, sınıflandırma işlemi yapılmıştır.

1.3.4 Filtreleme işleme (Medyan filtre)

Filtreleme işlemi görüntü işlemede ön işlem olarak çok sık kullanılan bir tekniktir. Görüntü işlemede kullanılabilecek birçok filtre arasından istenilen kriterlere en uygununun seçilmesi gerekmektedir. Düzenli istatistik filtreleri lineer olmayan uzaysal filtrelerdir ve bunların yanıtı filtre tarafından kapsanan alan içerisindeki imajın piksellerinin sıralanmasına dayanır ve daha sonra merkezi piksel değeri sıralama ile belirlenen sonuç değer ile değiştirilir. Bu kategoride en iyi bilinen örnek medyan filtredir ve isminden de anlaşılacağı üzere, bir pikselin değerini, o piksele komşu alanlardaki gri seviyelerin medyanı ile değiştirir. Pikselin orijinal değeri medyanın hesaplanmasına dahil edilir. Medyan filtreleri yaygın bir şekilde kullanılmaktadır. Çünkü, bazı rastgele gürültü türleri için bunlar mükemmel gürültü azaltma kabiliyeti göstermektedirler. Benzer boyuttaki lineer düzeltme filtrelerine göre çok daha az bulanıklığa neden olurlar. Medyan filtreler imaj üzerinde beyaz ve siyah noktalar halinde görünen ve tuz ve biber gürültüsü olarak da adlandırılan özellikle darbe gürültüsünün mevcut olduğu hallerde etkilidirler. Bir imajın içinde, bir noktada medyan filtrelemesi yapabilmek için, ilk önce ilişkili pikselleri ve komşu alandaki piksellerin bir düzene sokulması gerekir. Daha sonra medyan belirlenir ve bu değer piksele atanır. Bir komşu alanda birçok değer birbirinin aynı ise bütün aynı değerler gruplanır. Bu nedenle, medyan filtrelerinin başlıca fonksiyonu, belirgin gri seviyelere sahip olan noktaları daha fazla komşu alanlardaki gibi olmak üzere zorlamaktır [16]. Medyan filtreleme yöntemi, doğrusal olmayan bir işlemdir. Bu yöntem lineer filtreleme tekniklerinden daha seçici bir sonuç elde etmeye yaramaktadır. Medyan filtreleme, dürtü (impuls) gürültülerinin temizlenmesinde klasik yaklaşımdır. Örnek setlerinin medyanı olan $\{x(m)\}$ topluluğunun yarısının $x_{med}(m)$'den büyük ve yarısının $x_{med}(m)$'den küçük olduğu $x_{med}(m)$'nin üyesidir [17].

Medyan filtrede, lineer filtrelerdeki işlemlerin yanı sıra sonuç pikselin değeri, komşu piksellerin ortalama değerine bağlı olarak belirlenmiş olur. Matematiksel olarak ifade edilirse;

$$\hat{x}(m)=y_{med}(m)$$
$$\hat{x}(m)=\text{median}\left[y(m-k),...,y(m),...,y(m+k)\right] \tag{1.11}$$

$$\hat{x}(m)=\begin{cases}\int y(m), \text{eğer} \left|y(m)-y_{med}(m)\right|<kQ(m)\\ y_{med}(m), \text{diğer}\end{cases} \tag{1.12}$$

Denklem (1.1) ve (1.12) kullanılarak örnek bir görüntüye, medyan filtre matrisel formda aşağıdaki örnekteki gibi uygulanmıştır.

$$\begin{bmatrix}6 & 2 & 0\\ 3 & 97 & 4\\ 19 & 3 & 10\end{bmatrix}$$ Filtrelenmemiş piksel değerleri

Matris küçükten büyüğe doğru sıralanırsa; [0, 2, 3, 3, 4, 6, 10, 15, 97] medyan 4 olarak bulunur.

$$\begin{bmatrix}* & * & *\\ * & 4 & *\\ * & * & *\end{bmatrix}$$

Filtrelenmemiş merkez piksel değeri 97, medyan piksel değeri 4 ile değiştirilir.

Kullanılacak bilgiye göre dizi, matris veya görüntüye uygun olarak seçilebilecek çeşitli medyan maskeleri bulunmaktadır. Özetle, medyan filtre sayısal işaretlerde zorlayıcı uç değerlere sahip piksellerin, komşu piksellerin değerlerine yakınlaştırılması/benzetilmesi için kullanılır. Bu uç değerler, görüntü üzerindeki gürültüleri oluşturmaktadır. Aslında görüntü bilgisine ait olmayan ancak dış etkenlerle işaret içerisine yerleşen stokastik değerlerdir. Gürültü tipleri istatistiksel dağılımlarına göre sınıflandırılmıştır. Her bir gürültü için farklı bastırma yöntemleri geliştirilmiştir. Gürültü olarak tanımlanan işaretin özelliklerine ve çıkışta istenilen işaretin durumuna uygun filtre tasarımı sağlanır [18]. Bu çalışmada gerekli görülen yerlerde medyan filtre kullanılmıştır.

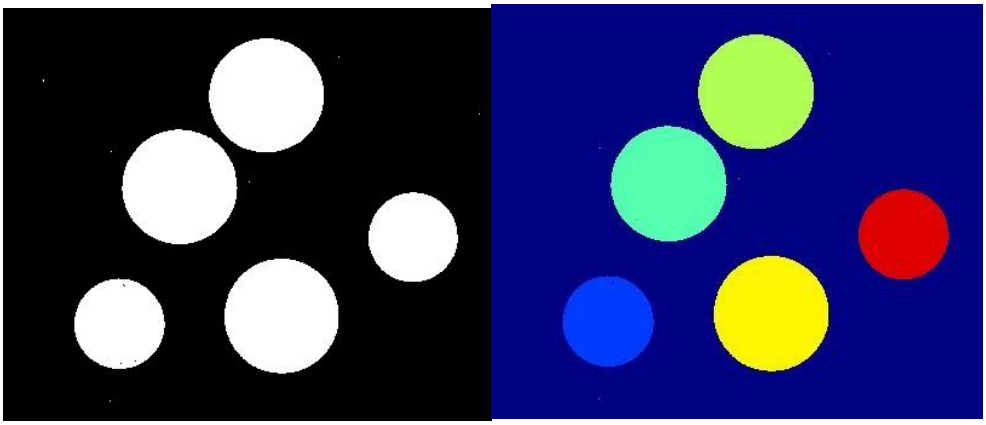
1.3.5 Bağlantılı bileşen etiketleme (Connected component labeling)

Görüntü işleme tekniklerine dayalı programlarda sıkça kullanılan bir yöntemdir. Türkçesi bağlantılı bileşen etiketleme yöntemi olan bu yöntem, ikili resimler üzerinde uygulanır ve üzerindeki bağımsız bileşenleri ayrı renkler ile etiketler. Böylece farklı bileşenlerin konumlarını tespit etmek kolaylaşır. Hatta konumları tespit edilen bileşenler birbirlerinin dikdörtgensel alanına girmiş olsalar bile ayrı ayrı başka görüntülere aktarılabilir [19].



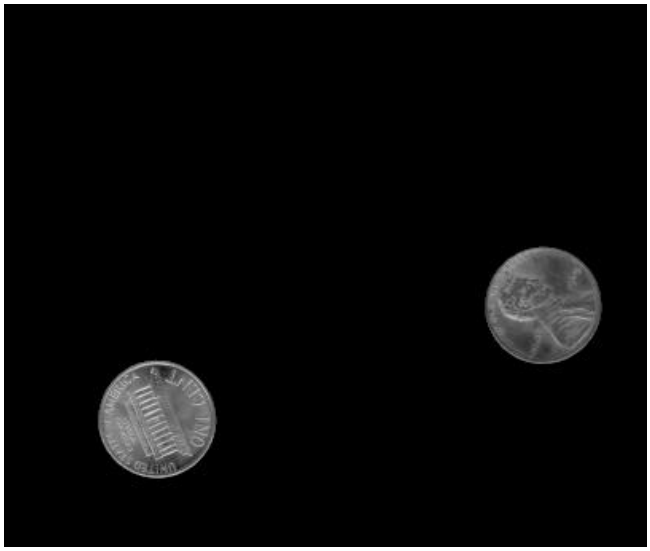
Şekil 1.11. Madeni paralar [20]

Şekil 1.11'de gösterilen görüntü, uygun eşikleme işlemine tabi tutulduktan sonra Şekil 1.12'deki siyah beyaz görüntü elde edilmektedir. Daha sonra her para farklı renge boyanmaktadır. Böylece her paranın farklı birer nesne olduğu anlaşılabilmektedir.



Şekil 1.12. Eşikleme uygulanmış imge ve tüm paraların farklı renge boyanması[20]

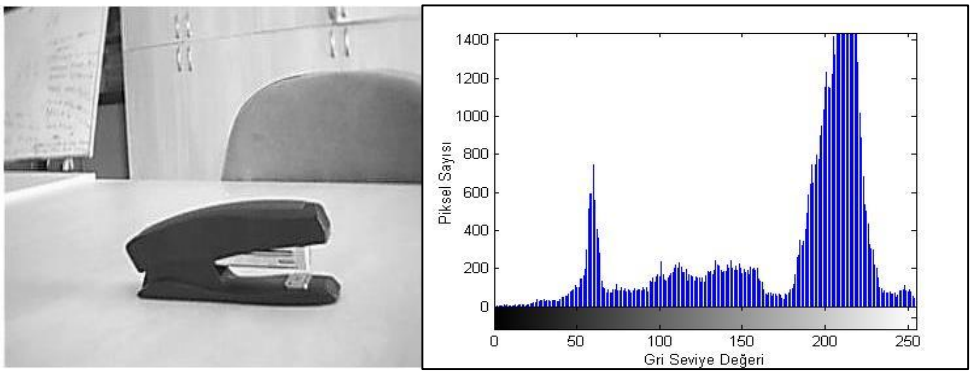
Her paranın kapladığı piksel hesaplanarak istenen para tek başına görüntüden çıkartılabilmektedir. Şekil 1.13'te istenen iki para çıkartılmış görüntü gösterilmektedir.



Şekil 1.13. İstenen iki para tüm imgeden çıkartılmış görüntü [20]

1.3.6 Histogram eşikleme

Gri seviyesi histogramı, görüntü içindeki her bir gri seviyesine karşılık gelen bölgeyi gösteren bir fonksiyondur. Apsis ekseni gri seviyesini gösterirken, ordinat ekseni ise meydana gelme sıklığını göstermektedir. Şekil 1.14'te görüntü ve ona karşılık gelen histogram grafiği gösterilmektedir.



Şekil 1.14. Görüntü ve histogram grafiği

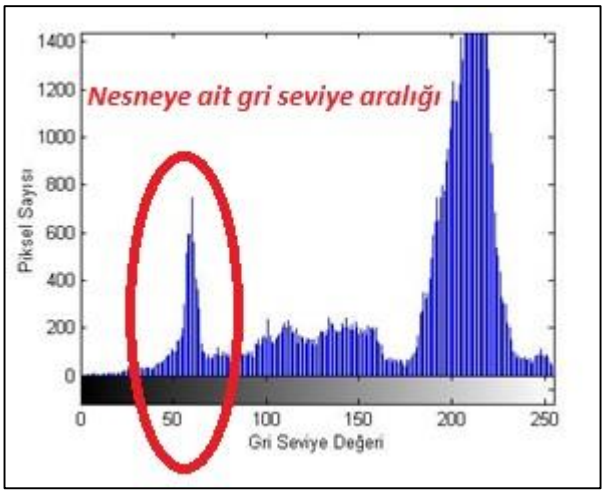
Histogram, sayısal bir resim içerisinde her renk değerinden kaç adet olduğunu gösteren grafiktir. Bu grafiğe bakılarak resmin parlaklık durumu ya da tonları hakkında bilgi sahibi olunabilir.

Bir görüntünün histogramı, o görüntü hakkında önemli bilgiler verir. En önemlileri aşağıda verilmektedir.

- Koyu (Karanlık) bir görüntünün histogram grafiğinin düşük gri seviye bölgesine yığılacağı açıktır.
- Parlak (Açık renk) düzgün bir görüntünün histogram grafiğinin büyük gri seviye bölgesine yığılacağı açıktır.
- Eğer histogram bir bölgeye yığılmış ise (yani gri seviye ekseninin belirli bir bölgesine) bu görüntünün kontrastı kötüdür denir.
- İyi kontraslı bir resmin histogram grafiği tüm gri seviye değerlerine eşit yayılmış olduğunu açıklar.
- Genelde küçük yayılım gösteren histogramların kontrastı düşüktür, daha geniş alana yayılmış olan histogramların kontrastı daha yüksektir.
- Aralığın aşağı sonunda sıkışmış olan histogram koyu bir görüntüye ait olurken, aynı aralığın yukarı sonunda gruplanmış olan histogram parlak bir görüntüye aittir [21].

Şekil 1.15'te histogram grafiğinde imgedeki piksel dağılımı gösterilmektedir. Görüntüde Beyaz (255)'e yakın piksel sayısının fazla olduğu görülmektedir. Yani gri tonlu piksel sayısı fazladır. Histogram grafiğinden de gri piksel sayısının fazla olduğu gözükmektedir. 200 civarında çokça piksel olduğu gözlenebilmektedir.

İmgedeki zımba nesnesinin hangi gri tonlardan oluştuğu histogram grafiğinden belirlenebilmektedir. Histogramdan nesnenin 50 civarındaki gri değerlerden oluştuğu gözlenmektedir. Şekil 1.15'te nesnenin ait olduğu piksel kümesi gösterilmektedir.



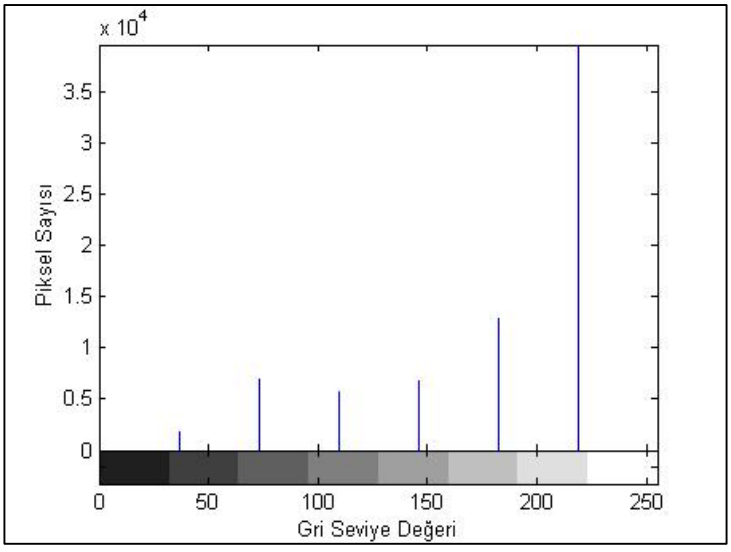
Şekil 1.15. Nesnenin ait olduğu piksel kümesi

İmhist fonksiyonu, ön ayarları yapılmaksızın, 256 gri seviyeli histogram vermektedir. Aşağıdaki fonksiyon ‘gray_imge’ imgesinin 256 gri seviyeli olarak piksel dağılımını göstermektedir.

```
h_3=imhist(gray_imge);
```

Histogram grafiği kümeleme yapılarak elde edilebilmektedir. İmhist fonksiyonu bu kullanıma izin vermektedir. Şekil 1.16’da 8 gri seviyeden oluşmuş histogram grafiği ve kullanılan kod verilmiştir.

```
h_3=imhist(gray_imge,8);
```

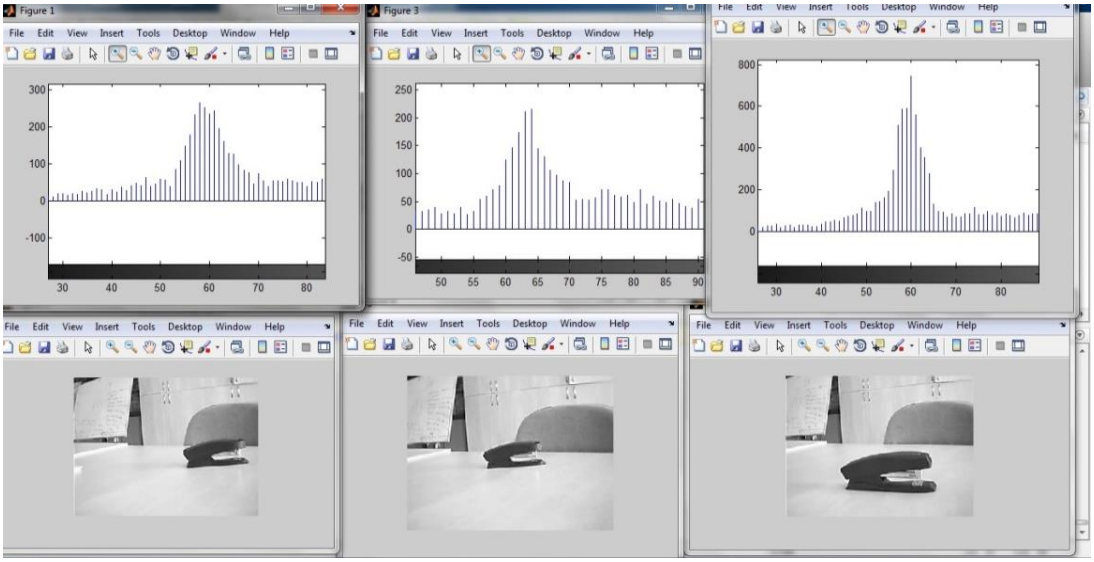


Şekil 1.16. 8 gri seviyeli histogram

8 gri seviyeli histogram 256 olan gri seviyesini 8'e indirger. Böylece imgede 8 gri seviyeli pikseller dağılımı gösterilir. Bu yaklaşım, yaklaşık olarak aynı gri seviyesine sahip pikselleri aynı kümeye atmaktadır.

Şekil 1.14'teki histogram yardımıyla nesne gri seviyesi belirlenebilmektedir. İmge için bu gri seviyeye ait piksel sayısının değişimi incelenerek nesnenin kameraya olan uzaklığı belirlenebilmektedir. 8 gri seviyeli histogramdan nesnenin ait olduğu küme daha kolay incelenebilmektedir. Böylece nesnenin uzaklığı daha kolay belirlenebilmektedir.

Şekil 1.17'den nesnenin imgede kapladığı yüzey alanı arttıkça ilgili alandaki piksel sayısının arttığı gösterilmiştir. Karşılaştırma üç farklı uzaklık için yapılmıştır.



Şekil 1.17. Üç farklı uzaklık için histogram karşılaştırması

Yaklaşık aynı mesafede olan nesne için ilgili bölge piksel sayısı 3000 civarlarındayken, kameraya daha yakın olan nesnenin ilgili bölge piksel sayı 6900 civarındadır.

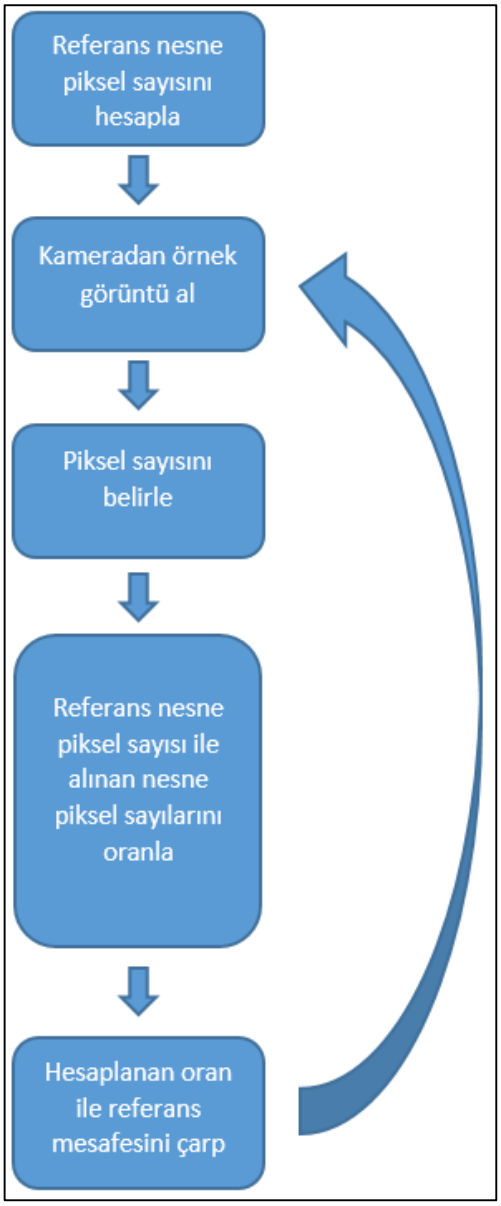
İlgili bölge: Nesnenin gri seviyesinin aralığıdır. 256 seviyeli histogramda 55-70 arasındadır.

Nesneye ait gri seviyesindeki piksel sayısı nesnenin kameraya uzaklığına göre değişmektedir. Böylece Histogram eşikleme yöntemi kullanılarak nesneye ait gri seviye kümesindeki değişime göre nesnenin kameraya olan uzaklığı

belirlenebilmektedir. Bölüm 4’te bu konu ile ilgili yapılan deneyler ve sonuçları verilmektedir.

2. SİSTEMİN GERÇEK ZAMANLI NESNE TAKİBİNDE KULLANILMASI

Bu tez çalışmasında, kamera sürekli çalıştırılıp saniyede bir görüntü alınarak nesnenin kameraya olan uzaklığı belirlenmektedir. Alınan görüntü Şekil 2.1'deki algoritmaya tabi tutularak nesnenin kameraya olan uzaklığı gerçek zamanlı olarak belirlenebilmektedir.



Şekil 2.1. Uzaklığın belirlenmesi için geliştirilen yöntemin akış diyagramı

İlk aşamada, referans görüntü sisteme tanıtılarak görüntüdeki nesnenin kaç pikselden oluştuğu belirlenmektedir. Aynı zamanda nesnenin kameraya olan uzaklığı da referans mesafesi olarak sisteme tanıtılmaktadır. Yani referans görüntüdeki nesnenin piksel sayısı ve kameraya olan uzaklığı bilinmektedir. Daha sonra kamera açılarak örnek görüntüler alınmaktadır. Alınan görüntüler, referans görüntünün tabi tutulduğu görüntü işleme işlemlerine tabi tutularak görüntüdeki nesnenin piksel sayıları belirlenmektedir. Böylece, piksel sayılarının oranı ve referans mesafesi kullanılarak, gerçek zamanlı olarak alınan her örnek görüntü için nesne uzaklığı belirlenebilmektedir. Kameranın açılmasında kullanılan kod parçası aşağıdaki gibidir.

```
vid=videoinput('winvideo',1);

preview(vid);
```

Kamera açıldıktan sonra sistem sonsuz döngüye sokulmaktadır. Sistem sonsuz döngüye girdikten sonra o anki kamera görüntüsünden örnek görüntü alınmaktadır. Örnek görüntü aşağıdaki kod parçası ile alınmaktadır. Yani kamera çalışırken aşağıdaki komut satırına gelindiğinde var olan görüntü frame değişkene atılmaktadır.

```
frame = getsnapshot(vid);
```

‘Frame’ değişkenindeki görüntü, nesne piksel sayısını belirleme algoritmasına gönderilir. Piksel sayısı belirlendikten sonra referans görüntü piksel sayısı ile oranlanarak referans mesafesi ile çarpılır. Çarpımın sonucu nesnenin kameraya olan uzaklığını vermektedir. Böylece alınan görüntüde nesne tespiti ve nesnenin kameraya olan uzaklığı belirlenmektedir. Sistem sonsuz döngü ile, gerçek zamanlı olarak sürekli görüntü alıp nesnenin kameraya olan uzaklığını hesaplamaktadır. Gerçek zamanlı çalışma Şekil 2.2’deki referans görüntü üzerinden yapılmıştır.



Şekil 2.2. Referans görüntü

Uzaklık hesabı nesnenin tespiti ve piksellerinin sayısına göre yapılmıştır. Uzaklık hesaplama işlemi, yakın nesnelerin uzak nesnelere göre görüntüde daha fazla piksel kaplaması mantığına göre yapılmıştır. Bu bağlamda, Şekil 2.1'deki algoritma geliştirilerek daha önceden sisteme tanıtılmış bir nesnenin gerçek zamanlı olarak kameraya olan uzaklığı belirlenmiştir.

Çalışmada, ilk olarak Şekil 2.2'de görünen mavi nesnenin tespiti için uygun renk kanalı ve uygun eşik belirlenmiştir. Görüntü HSV uzayına geçirildikten sonra S kanalı kullanılarak uygun eşik altında mavi nesne tespiti başarıyla gerçekleştirilmiştir.

Mavi nesne tespit edildikten sonra, nesnenin kaç pikselden oluştuğunu daha rahat belirleyebilmek için, arka taraflar söndürülerek tüm görüntüde nesne sadece beyaz piksellerle ifade edilmektedir. Ayrıca, oluşabilecek gürültüler de morfolojik işlemlerle giderilerek nesne daha belirgin hale getirilmektedir. Morfolojik işlemler kullanılarak (açma-kapama işlemleri) nesnelerin belirginleşmesi sağlanır. Matematiksel morfoloji temel küme işlemlerine dayanan, imgedeki sınırlar (borders), iskelet (skeleton) gibi yapıların tanımlanması ve çıkartılması, gürültü giderimi, bölütleme gibi uygulamalar için gerekli bir araçtır. İmge işlemede genellikle, morfolojik süzgeçleme, inceltme (thinning), budama (pruning) gibi ön/son işlem olarak da sıkça kullanılırlar. Gri tonlu imgeler üzerinde de yapılabileceği gibi, genellikle ikili imgeler üzerinde yapılan işlemlerdir. Gürültüler piksel sayısının yanlış hesaplanmasına sebebiyet vermektedir. Bu da nesnenin uzaklığının yanlış hesaplanmasına sebep olmaktadır.



Şekil 2.3. Eşikleme sonucunda elde edilen görüntü

Şekil 2.3'teki görüntünün beyaz pikselleri sayılarak, referans görüntüdeki nesnenin kaç pikselden oluştuğu belirlenmiştir. Çalışmada "referans mesafesi 50cm." olarak alınmıştır.

Şekil 2.3'te gösterildiği gibi, eşikleme sonucu mavi nesne beyaz pikseller olarak ifade edilmiştir. Görüntüdeki tüm pikseller gezilerek beyaz olan pikseller sayılmaktadır. İşlem sonucunda, referans görüntünün piksel sayısı "ref_pixel =4315" olarak bulunmuştur.



Şekil 2.4. Gerçek zamanlı işlem sırasında alınan kamera görüntüsü



Şekil 2.5. Gerçek zamanlı görüntünün siyah-beyaz formu

Gerçek zamanlı olarak alınan ikinci görüntüdeki piksel sayısı, yine yukarıda kulanılan işlemler kullanılarak sayılmaktadır. Kod parçasının çalışması sonucunda Şekil 2.4'teki görüntü için elde edilen piksel sayısı "hesaplanan_pixel_sayisi = 5909" olarak bulunmuştur. Şekil 2.5'te kullanılan nesnenin siyah-beyaz görüntüsü görülmektedir.

Böylece elde tek bilinmeyenli iki ilişki bulunmaktadır. Bilinmeyen mesafeyi bulabilmek için Denklem (2.1) kurulmuştur.

referans mesafesi = 50 cm. ref_pixel = 4315 (2.1)

bilinmeyen nesne uzaklığı hesaplanan_pixel_sayisi = 5909

Nesne yaklaştıkça piksel sayısı artacaktır. Nesne uzaklığını belirlemek için ters orantı kurulmuştur. O halde;

bilinmeyen nesne uzaklığı = (ref_piksel÷hesaplanan_piksel_sayisi)×ref_mesafe (2.2)

Yani; bilinmeyen nesne uzaklığı = (4315÷5909)×50cm.

bilinmeyen nesne uzaklığı = 0,7302×50cm. = 36,5121cm.

Sonuç olarak, gerçek zamanlı ikinci görüntüdeki nesnenin kameraya olan uzaklığı, referans görüntüdeki piksel sayısı ve ikinci görüntüdeki piksel sayısı oranının referans mesafe ile çarpımıyla Denklem (2.2) kullanılarak, "36,5121" cm. olarak bulunmuştur.

3. DENEYSEL SONUÇLAR

Önceki bölümde, gerçek zamanlı olarak renk tabanlı nesne tespiti ve takibi başarıyla gerçekleştirilmiştir. Nesne tespit edilip gerçek zamanlı mesafe hesabı yapılmıştır. Bu bölümde ise, Bölüm 1'de teorik olarak anlatılan yöntemlerin uygulanabilirliği incelenmiştir. Elde edilen sonuçlar karşılaştırılarak tek kamerayla nesne uzaklık ölçümü için en optimal yöntem geliştirilmiştir.

3.1. HSV Uzayında Eşikleme Kullanılarak Yapılan Ölçüm

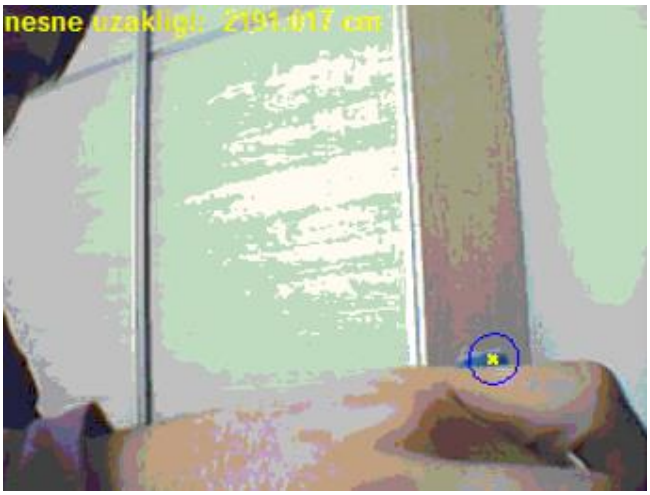
HSV uzayında eşikleme yapılarak çeşitli nesneler üzerinde ve çeşitli uzaklıklarda mesafe ölçümü yapılmıştır.

3.1.1. Silindirik nesnelerin HSV uzayında eşikleme kullanılarak 0-1m. aralığında uzaklık ölçümü

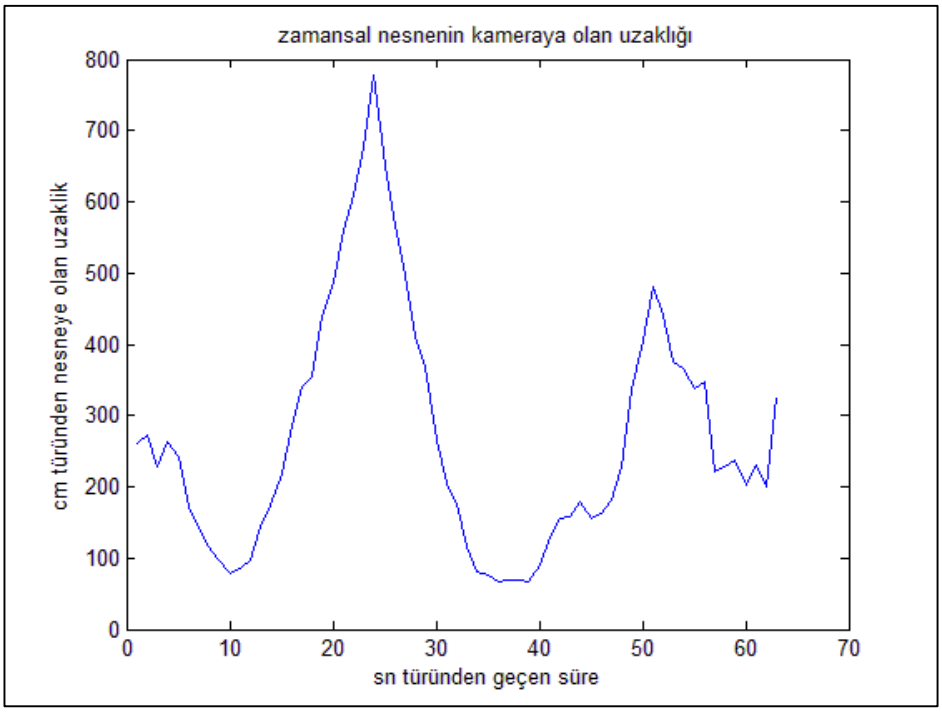
Sistem piksel sayısına göre mesafe belirlediği için, aynı uzaklıkta olup Şekil 3.1'deki gibi silindirik bir cismin tabanı, Şekil 3.2'deki gibi kameraya çevrildiğinde nesne uzaktaymış gibi algılanmaktadır. Çünkü piksel sayısı silindirin yüzeyinden daha az olarak hesaplanmaktadır. Mesafe piksel sayıları oranı kullanılarak belirlendiği için, otomatik olarak nesne uzaktaymış gibi algılanmaktadır. Şekil 3.3'te gerçek zamanlı olarak nesnenin kameraya olan uzaklığı gösterilmiştir.



Şekil 3.1. Silindirik yüzey gösterildiğinde ölçülen uzaklık: 195,6cm.



Şekil 3.2. Taban yüzey gösterildiğinde ölçülen uzaklık: 2193,4cm.

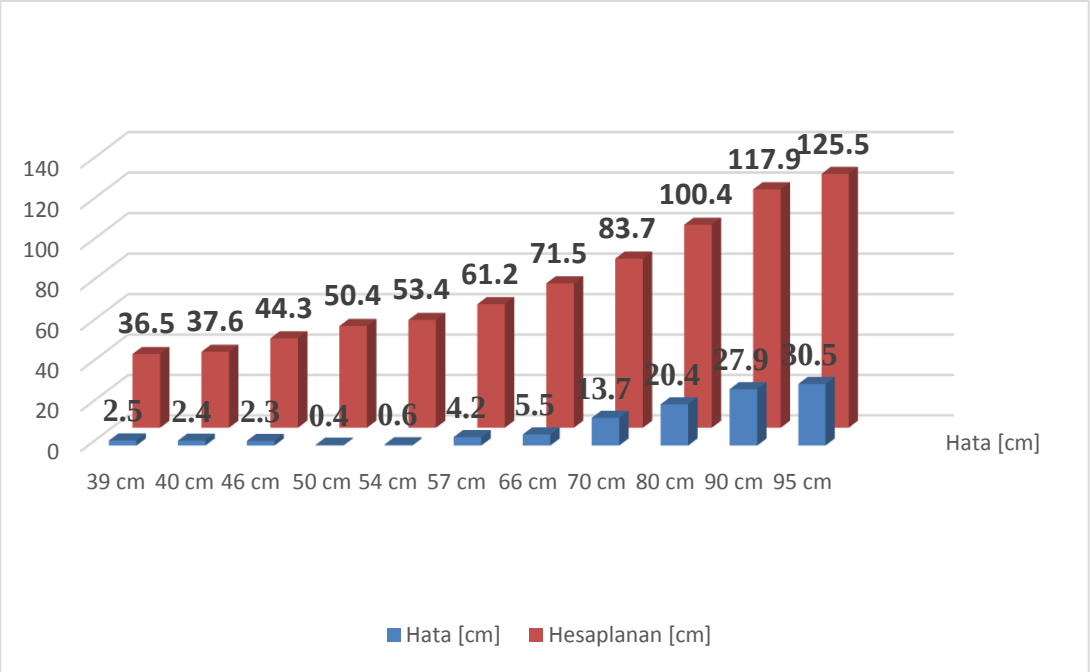


Şekil 3.3. Gerçek zamanlı olarak nesnenin kameraya olan uzaklık ölçümü

Nesnenin kameraya olan uzaklığının doğru olarak hesaplanıp hesaplanmadığı incelenmiştir. Bu inceleme, nesne belirli mesafelere koyulup sistemin verdiği cevap gözlenerek yapılmıştır. Tablo 3.1'de elde edilen sonuçlar gösterilmektedir.

Tablo 3.1. Deneyler uzaklık ölçümleri ve oluşan hatalar

Gerçek Mesafe [cm.]	Hesaplanan [cm.]	Hata [cm.]
39	36,5	2,5
40	37,6	2,4
46	44,3	2,3
50	50,4	0,4
54	53,4	0,6
57	61,2	4,2
66	71,5	5,5
70	83,7	13,7
80	100,4	20,4
90	117,9	27,9
95	125,5	30,5



Şekil 3.4. Hata oranları ve ölçümler

Şekil 3.4'ten de görüldüğü gibi, nesne uzaklaştıkça sistemin verdiği hata artmaktadır. Sonuç olarak, küçük nesneler için kamerayla mesafe hesabı yapabilmek için nesnenin kameraya 100cm.'den uzak olmaması gerekmektedir. Aksi takdirde elde edilen uzaklık bilgileri kabul edilemez hatalar içerecektir. Çözüm olarak 100cm.'den uzak nesnelerde mesafe hesabı yapabilmek için daha büyük nesneler ve daha iyi ışıklandırılmış ortamlar tercih edilmelidir. Bu şekilde yapılacak ölçümler daha sağlıklı sonuçlar verecektir.

3.1.2. Büyük nesnelerin HSV uzayında eşikleme kullanılarak 1m.-5m. aralığında ölçümü

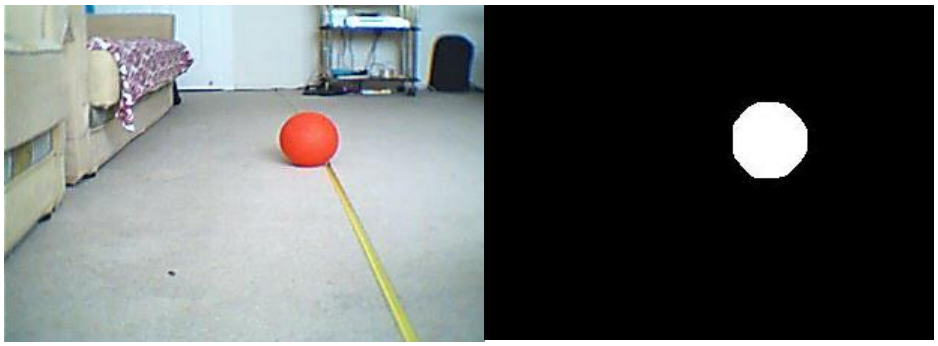
Referans görüntü 1m. iken nesnenin kameraya olan uzaklığı ölçülmüştür. Ölçümler 2m.,3m.,4m. ve 5m. mesafe için yapılmıştır. Yapılan ölçümlerde Şekil 2.1'deki algoritma kullanılmıştır.

3.1.2.1. Büyük balon referans görüntü 1m. iken ölçüm

1m.'deki balon referans görüntü olarak kullanılmıştır. Elde edilen sonuçlar Tablo 3.2'de gösterilmiştir.



Şekil 3.5. Referans görüntü



Şekil 3.6. 2m.'deki balon görüntüsü



Şekil 3.7. 3m.'deki balon görüntüsü



Şekil 3.8. 4m.'deki balon görüntüsü

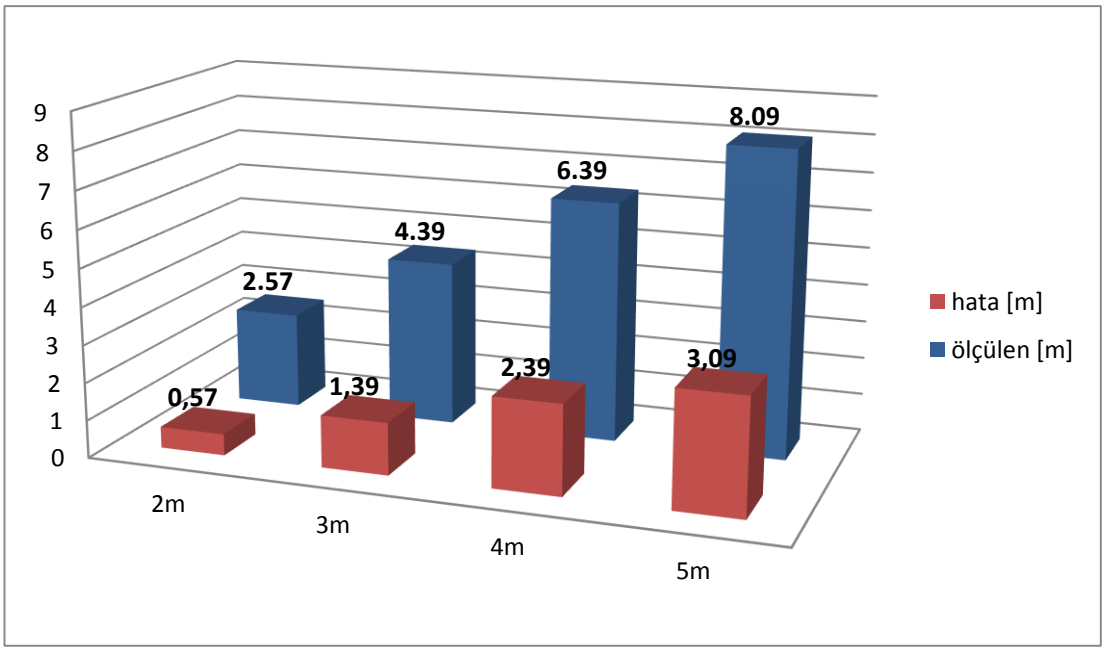


Şekil 3.9. 5m.'deki balon görüntüsü

Şekil 3.5'te gösterilen görüntü kameraya 1m. uzaklıktadır. Bu görüntü, geliştirilen algoritma için referans olarak kullanılmıştır. 1m. uzaklıktaki referans nesnenin görüntüde kapladığı piksel sayısı 6015 olarak belirlenmiştir. Şekil 3.6, Şekil 3.7, Şekil 3.8 ve Şekil 3.9'da gösterilen görüntülerin kameraya olan uzaklıkları, ölçülen uzaklıklar ve piksel değerleri Tablo 3.2'de gösterilmektedir.

Tablo 3.2. Referans görüntü 1m. iken nesne uzaklık ölçümü

Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
2	2334	2,57
3	1368	4,39
4	941	6,39
5	743	8,09

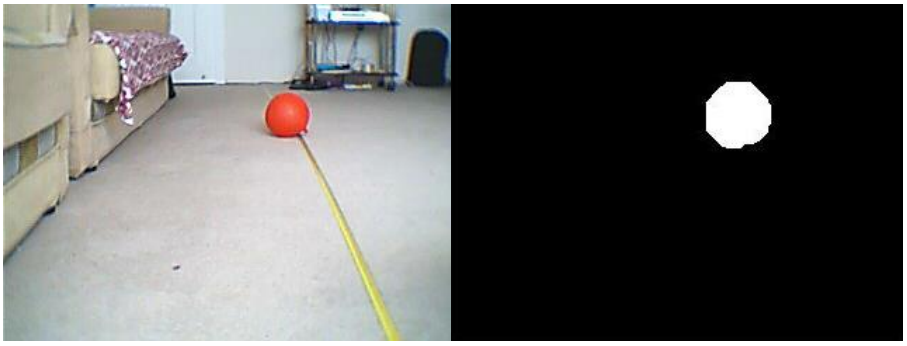


Şekil 3.10. Referans 1m. iken alınan sonuçlar

Şekil 3.10'daki grafikten referans görüntüden uzaklaşıldıkça hata oranın arttığı gözlemlenmektedir. Hata oranını azaltmak için referans görüntü 2,5m. uzaklıktayken ölçümler tekrarlanmıştır.

3.1.2.2. Büyük balon referans görüntü 2,5m. iken ölçüm

Referans görüntü 2,5m. iken nesnenin kameraya olan uzaklığı ölçülmüştür. Ölçümler 2m.,3m.,4m. ve 5m. mesafe için yapılmıştır. Elde edilen sonuçlar Tablo 3.3'te gösterilmiştir.

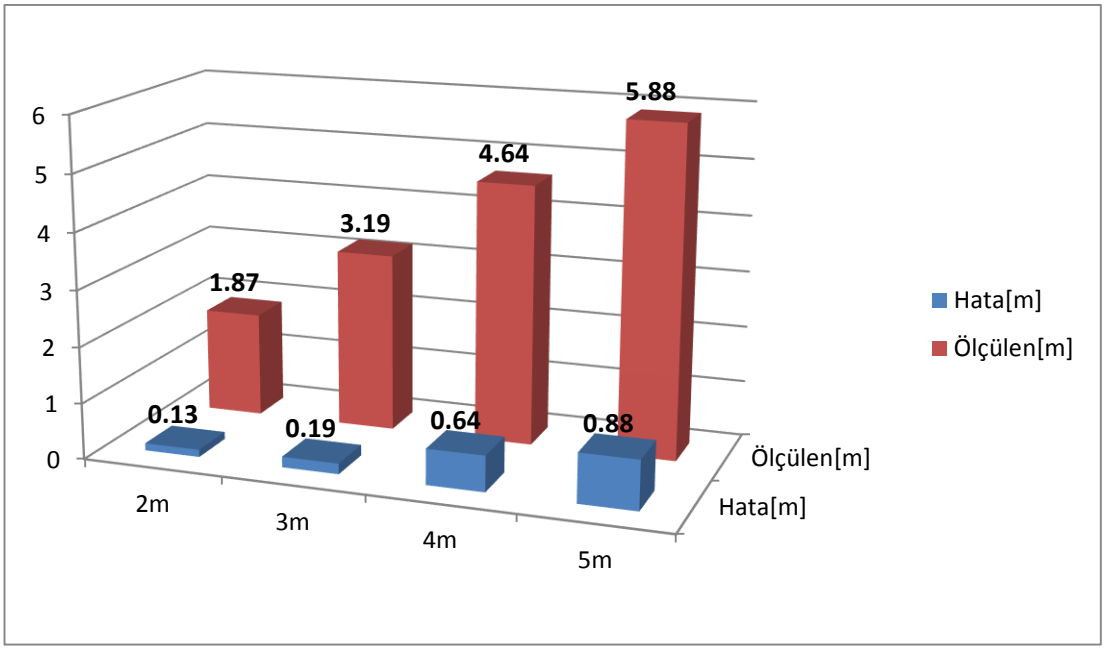


Şekil 3.11. Referans görüntü

Şekil 3.11'de gösterilen görüntü kameraya 2,5m. uzaklıktadır. Bu görüntü geliştirilen algoritma için referans olarak kullanılmıştır. 2,5m. uzaklıktaki referans nesnenin görüntüde kapladığı piksel sayısı 1749 olarak belirlenmiştir.

Tablo 3.3. Referans görüntü 2,5m. iken nesne uzaklık ölçümü

Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
2	2334	1,87
3	1368	3,19
4	941	4,64
5	743	5,88



Şekil 3.12. Referans görüntü 2,5m. iken ölçüm sonuçları

Şekil 3.10 ve Şekil 3.12'den gözlendiği gibi referans mesafesi 2,5m. olduğu zaman çok daha iyi sonuçlar alınmıştır. Referans mesafesinin yaklaşık 1m. civarında daha sağlıklı sonuçlar alınmaktadır. Nesne referans mesafesinden uzaklaştıkça hata oranı artmaktadır.

3.1.2.3. Defter referans görüntü 1,5m. iken ölçüm

Aynı işlemler kare bir nesne ile yapılmıştır. Farklı nesneler ve farklı referans mesafeleri kullanılarak ölçüm sonuçları incelenmiştir. Referans mesafesi 1,5m. olan kırmızı defter için, referans piksel 'ref_piksel=4588' olarak bulunmuştur. Referans görüntü olarak Şekil 3.13'teki görüntü kullanılmıştır.



Şekil 3.13. Referans görüntü

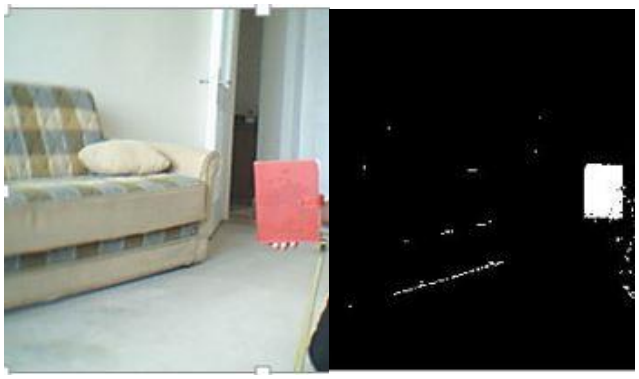
Kameraya olan uzaklığın artması, nesnenin ve ortamın değişmesi nedeniyle tekrardan eşikleme yapılmıştır.



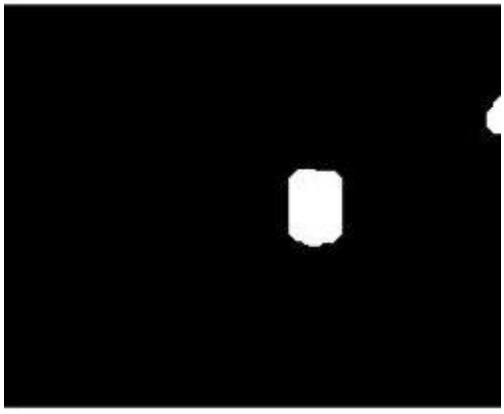
Şekil 3.14. İlk eşikleme ve çeşitli işlemler sonucunda elde edilen görüntü

Şekil 3.14'ten de gözlendiği gibi elde edilen görüntü oldukça gürültülüdür. Aslında tespit edilmek istenen görüntü Şekil 3.13'te gözüken kırmızı defterdir. Şekil 3.14 üzerinde olabildiğince morfolojik işlemlerle gürültüler giderilmeye, kırmızı defter bölgesi belirlenmeye çalışılmıştır. Sonuç olarak, referans mesafesi 1,5m. olan kırmızı defter için, referans piksel 'ref_piksel=4588' olarak bulunmuştur.

Ölçümler 2m.-2,3m.-2,5m.–3m.-3,5m. için yapılmıştır.



Şekil 3.15. 2m.'deki görüntü ve eşikleme sonucu elde edilen görüntü



Şekil 3.16. Morfolojik işlemler sonucunda elde edilmiş görüntü

Görüntü üzerinde detaylı çalışılmasına karşın en uygun eşik değerinde bile oldukça gürültülü bir sonuç elde edilmiştir. Şekil 3.15'te eşikleme sonucunda elde edilmiş gürültülü görüntü gösterilmektedir. Gürültünün giderilmesi için Şekil 3.15'e birden fazla morfolojik işlem uygulanmıştır. Morfolojik işlemler sonucunda kırmızı defter bölgesi ancak Şekil 3.16'daki gibi elde edilebilmiştir. Görüntüde kırmızı defterin olmadığı piksellerde aranan nesneymiş gibi belirlenmiştir. Bu durum piksel sayının farklı çıkmasına dolayısıyla yanlış mesafe hesaplanmasına sebebiyet verecektir. Şekil 3.16 üzerinden yapılan piksel sayma sonucunda 2m.'deki kırmızı defter için piksel sayısı 'hesaplanan_pixel_sayisi=2273' olarak bulunmuştur.

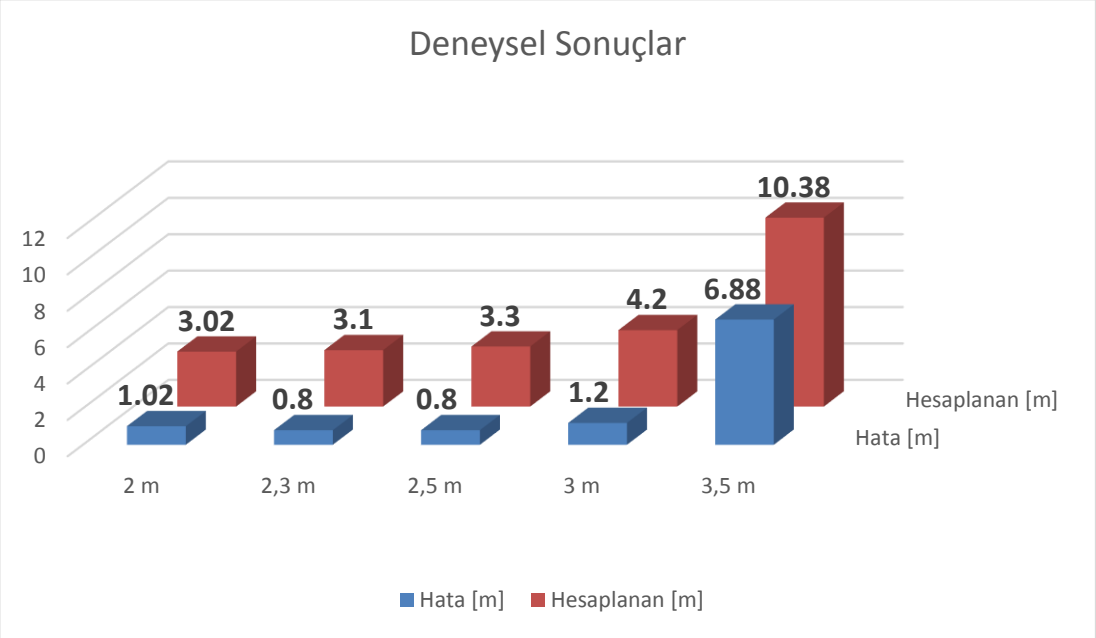
bilinmeyen nesne uzaklığı = (ref_pixel ÷ hesaplanan_pixelsayisi)×referans mesafe

bilinmeyen nesne uzaklığı = (4588÷2273)×1,50m. = 3,02m.

2m.'de olan görüntü Denklem (2.2) kullanılarak, 3,02m. olarak ölçülmüştür. 1,02m. hata oluşmaktadır. Bunun gibi farklı mesafelerde görüntüler incelenmiştir. Tablo 3.4'te, yapılan testlerin sonuçları verilmiştir. Şekil 3.17'de hesaplanan değerler ve hata oranları gösterilmektedir.

Tablo 3.4. Referans 1,5m. iken hesaplanan mesafeler ve hata değerleri

Gerçek Mesafe [m.]	Hesaplanan [m.]	Hata [m.]
2	3,02	1,02
2,3	3,1	0,8
2,5	3,3	0,8
3	4,2	1,2
3,5	10,38	6,88



Şekil 3.17. Deneysel sonuç grafiği

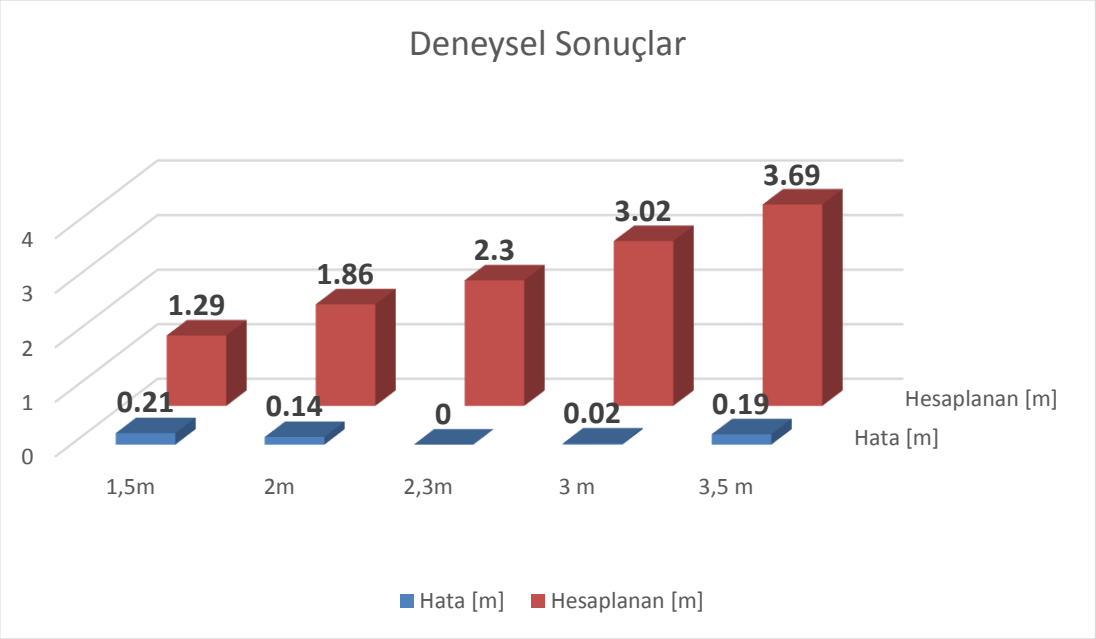
Özellikle 3,5m.'de ve daha sonraki mesafelerde gürültülerin artması sebebiyle kırmızı defterin mesafesi iyi kestirilememektedir. Yapılan işlemler sonucunda gürültü giderilse bile, defteri temsil eden piksel sayısı morfolojik işlemler nedeniyle sağlıklı belirlenememektedir.

3.1.2.4. Defter referans görüntü 2,5m. iken ölçüm

Referans görüntü değiştirilince hata oranları da değişmektedir. Balon örneğinde olduğu gibi defterde de daha iyi sonuçlar elde edebilmek için 2,5m.'de referans görüntü kullanılmıştır. Tablo 3.5'te yapılan testlerin sonuçları verilmiştir.

Tablo 3.5. Referans 2,5m. iken hesaplanan mesafeler ve hata değerleri

Gerçek Mesafe [m.]	Hesaplanan [m.]	Hata [m.]
1,5	1,29	0,21
2	1,86	0,14
2,3	2,30	0
3	3,02	0,02
3,5	3,69	0,19



Şekil 3.18. Referans 2,5m. iken deneysel sonuç grafiği

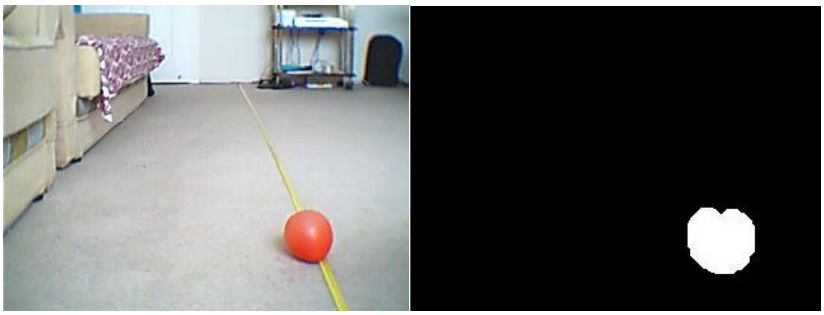
Görüntü HSV uzayında işlenirken nesne belirlenmesi daha zor olmuştur. Nesnenin daha kolay belirlenebilmesi için yeni eşik belirlenmiştir. Şekil 3.17 ve Şekil 3.18'den gözleneceği gibi referans mesafesi 2,5m. olduğu zaman çok daha iyi sonuçlar alınmıştır. Referans mesafesinin yaklaşık 1m. civarında daha sağlıklı sonuçlar alınmaktadır. Nesne referans mesafesinden uzaklaştıkça hata oranı artmaktadır. Ölçüm sonuçlarını iyileştirmek için sınıflandırma yapılarak ölçüm yapılmıştır.

3.1.3. Küçük nesnelerin HSV uzayında eşikleme kullanılarak 1m.-5m. aralığında ölçümü

İlk incelemeler büyük nesneler için yapılmıştır. Bu kısımda aynı yöntemin daha küçük nesneler için nasıl sonuçlar verdiği incelenmiştir. 1m.-5m. aralığında balondan 1/3 oranında küçük nesneler incelenmiştir. Referans görüntü 1m. ve 2,5m. için yöntem test edilmiştir.

3.1.3.1. Küçük balon referans görüntü 1m. iken ölçüm

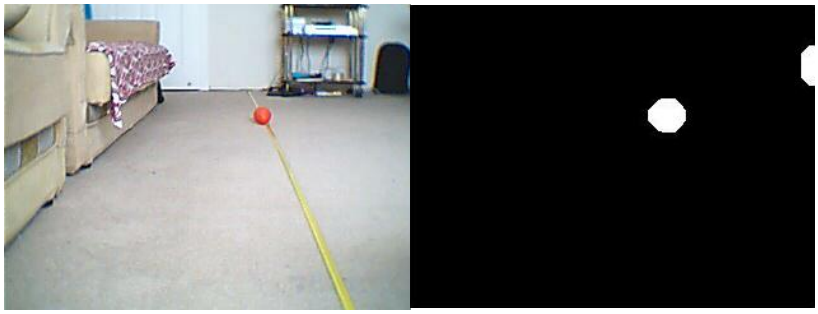
1m.'deki küçük balon referans görüntü olarak kullanılmıştır. Elde edilen sonuçlar Tablo 3.6'da gösterilmiştir.



Şekil 3.19. Küçük nesne için 1m.' deki referans görüntü

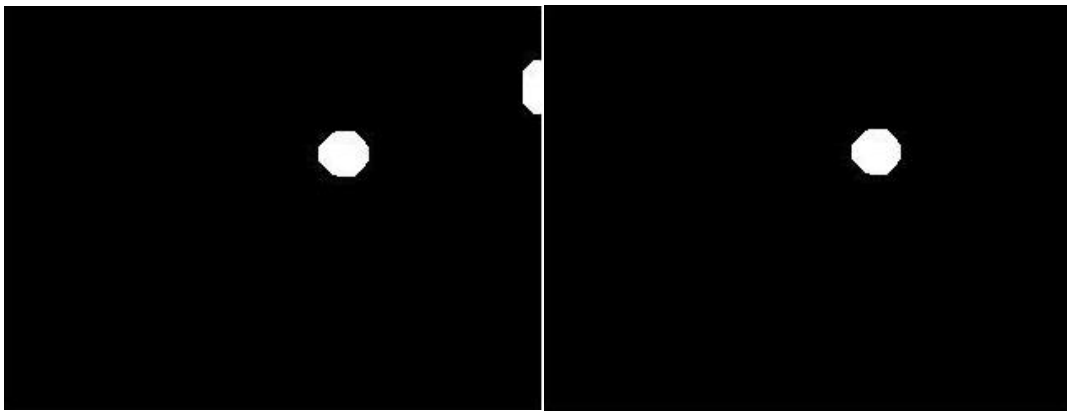


Şekil 3.20. Küçük nesne için 2m.'deki görüntü



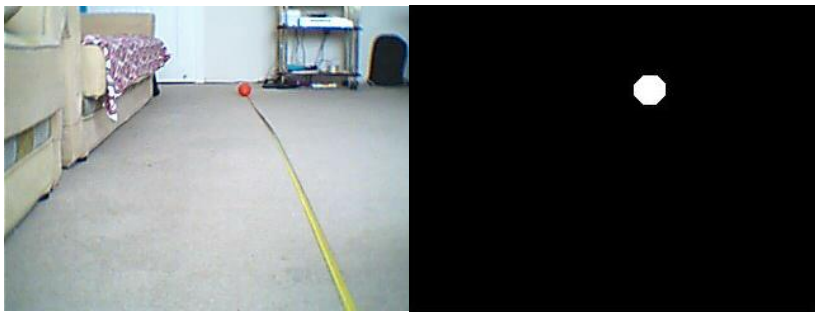
Şekil 3.21. Küçük nesne için 3m.'deki görüntü

3m.'deki görüntünün HSV uzayında eşiklenmesi sonucunda Şekil 3.21'deki gibi balon haricinde başka bir nesne olan görüntü de elde edilmiştir. Bu görüntüden asıl aradığımız nesneyi belirleyebilmek için 'Connected Component Labeling' yöntemi kullanılmıştır. Yöntem "daha çok piksel sayısına sahip alan aranan nesnedir diğer bulunan alanlar gürültüdür." yorumu yaparak aranan nesneyi belirlemektedir. Connected Component Labeling kullanılarak elde edilen görüntü Şekil 3.22'de gösterilmektedir.

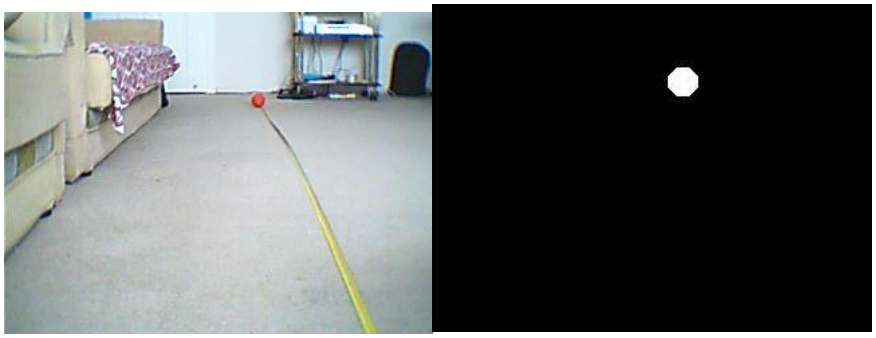


Şekil 3.22. Connected Component Labeling yöntemi sonucunda elde edilen görüntü

İlk aşamada Connected Component Labeling yöntemi uygulanmadan önce hesaplanan piksel 951'idi. Yöntem uygulanıp asıl olan nesne belirlendikten sonra hesaplanan piksel 645'tir. Şekil 3.22'de bu piksel farkını göstermektedir.



Şekil 3.23. Küçük nesne için 4m.'deki görüntü

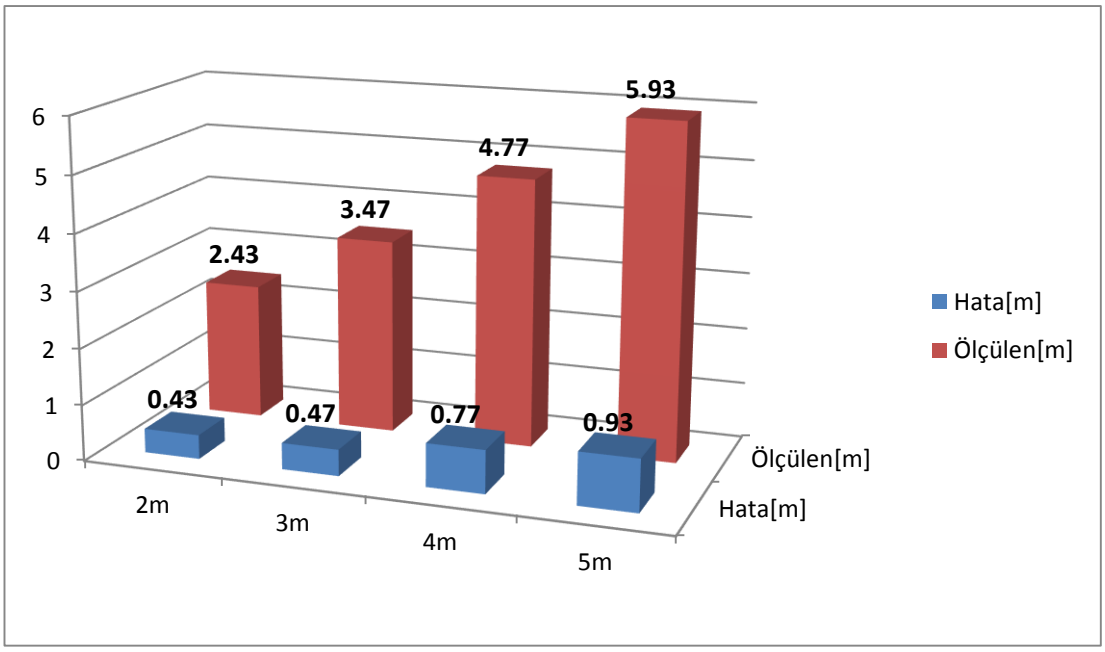


Şekil 3.24. Küçük nesne için 5m.'deki görüntü

Şekil 3.19'da gösterilen görüntü kameraya 1m. uzaklıktadır. Bu görüntü geliştirilen algoritma için referans olarak kullanılmıştır. 1m. uzaklıktaki referans nesnenin görüntüde kapladığı piksel sayısı 2244 olarak belirlenmiştir. Şekil 3.20, Şekil 3.21, Şekil 3.23 ve Şekil 3.24'teki görüntülerin kameraya olan uzaklıkları referans görüntüye göre ölçülmüştür. Sonuçlar Tablo 3.6'da verilmiştir. Ölçülen uzaklıkların hata oranları Şekil 3.25'te gösterilmektedir.

Tablo 3.6. Referans görüntü 1m. iken nesne uzaklık ölçümü

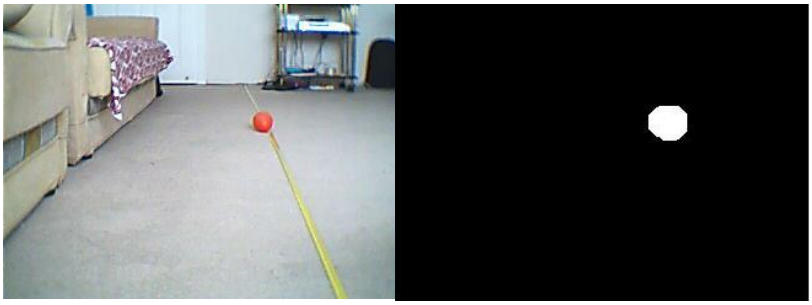
Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
2	923	2,43
3	645	3,47
4	470	4,77
5	378	5,93



Şekil 3.25. Referans görüntü 1m. iken ölçüm sonuçları

3.1.3.2. Küçük balon referans görüntü 2,5m. iken ölçüm

Referans görüntü 2,5m. iken küçük balonun kameraya olan uzaklığı ölçülmüştür. Ölçümler 2m., 3m., 4m. ve 5m. mesafe için yapılmıştır. Elde edilen sonuçlar Tablo 3.7'de gösterilmiştir.

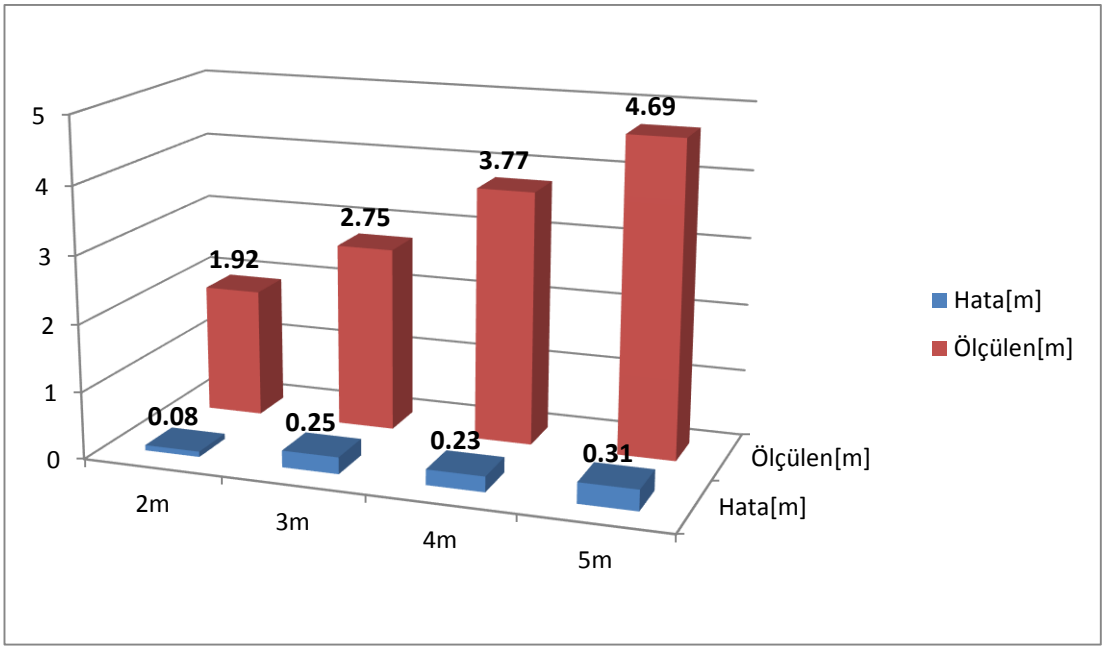


Şekil 3.26. Küçük nesne için 2,5m.'deki görüntü

Şekil 3.26'da gösterilen görüntü kameraya 2,5m. uzaklıktadır. Bu görüntü, geliştirilen algoritma için referans olarak kullanılmıştır. 2,5m. uzaklıktaki referans nesnenin görüntüde kapladığı piksel sayısı 710 olarak belirlenmiştir.

Tablo 3.7. Referans görüntü 2,5m. iken nesne uzaklık ölçümü

Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
2	923	1,92
3	645	2,75
4	470	3,77
5	378	4,69



Şekil 3.27. Referans görüntü 2,5m. iken ölçüm sonuçları

Şekil 3.25 ve Şekil 3.27'den görüldüğü gibi referans mesafesi 2,5m. olduğu zaman çok daha iyi sonuçlar alınmıştır. Referans mesafesinin yaklaşık 1m. civarında daha sağlıklı sonuçlar alınmaktadır. Nesne referans mesafesinden uzaklaştıkça hata oranı artmaktadır.

3.1.4. 5m., 7,5m., 10m., 12,5m., 15m. için farklı nesneler kullanılarak yapılan ölçüm

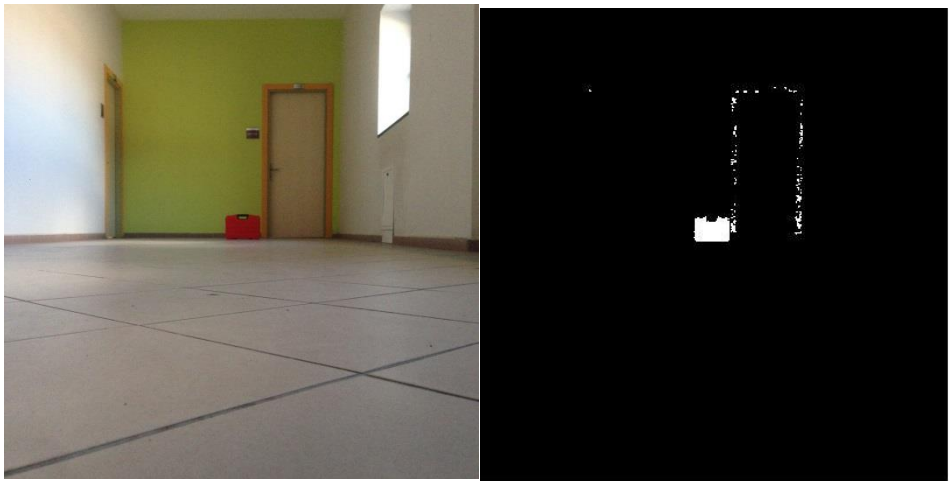
Bu bölümde farklı nesneler için 5m.-15m. arası uzaklıklar için ölçümler yapılmıştır.

3.1.4.1. Kırmızı çanta referans görüntü 5m. iken ölçüm

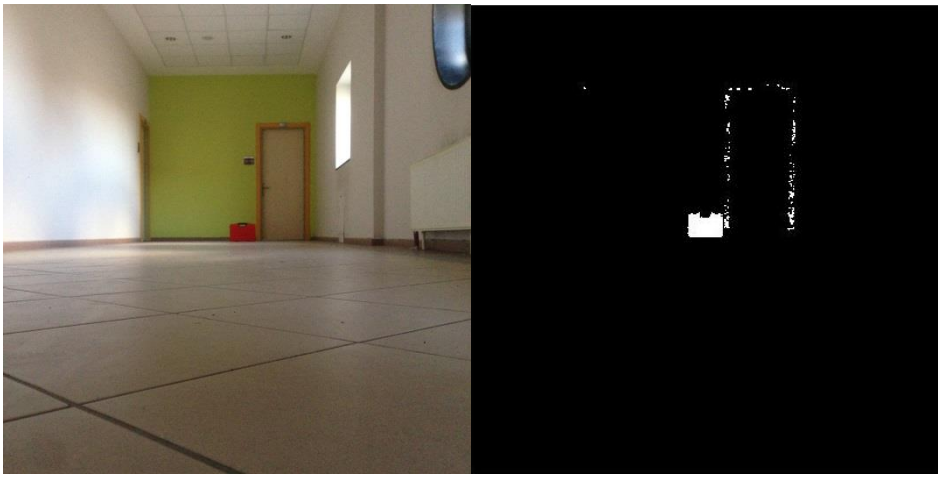
Referans görüntü olarak, kameraya 5m. uzaklıkta olan Şekil 3.28'deki görüntü kullanılmıştır. Şekil 3.29, Şekil 3.30, Şekil 3.31 ve Şekil 3.32'deki görüntülerin kameraya olan uzaklıkları ölçülmüştür. Elde edilen sonuçlar Tablo 3.8'de gösterilmiştir. Şekil 3.33'te hata analiz grafiği görülmektedir.



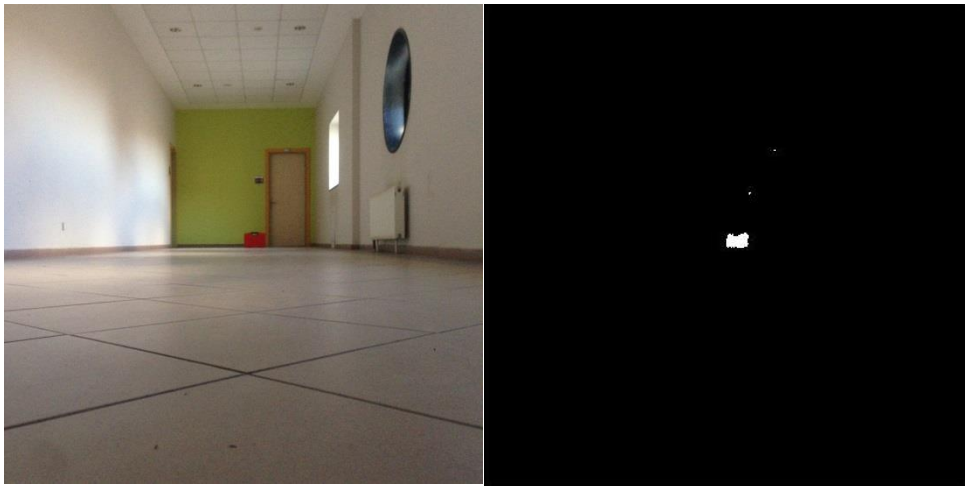
Şekil 3.28. Referans görüntü 5m.'de çanta



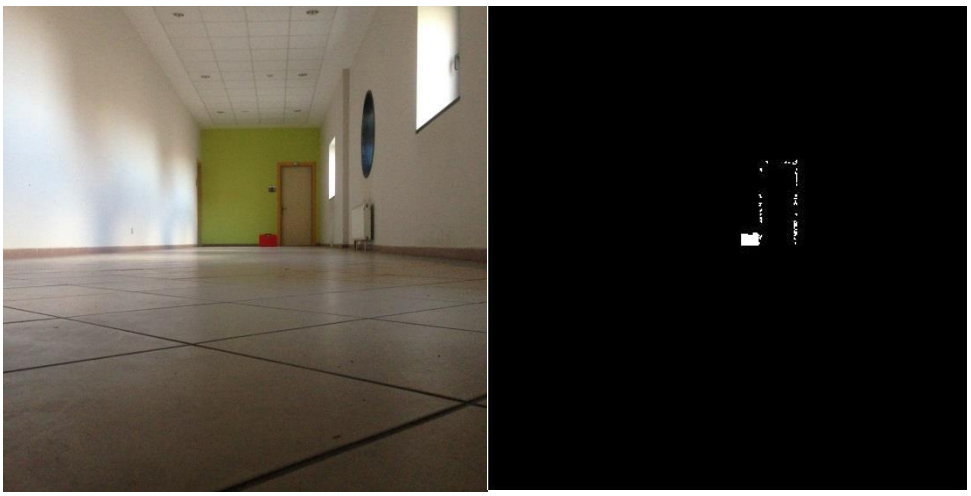
Şekil 3.29. Çanta için 7,5m.'deki görüntü



Şekil 3.30. Çanta için 10m.'deki görüntü



Şekil 3.31. Çanta için 12,5m.'deki görüntü

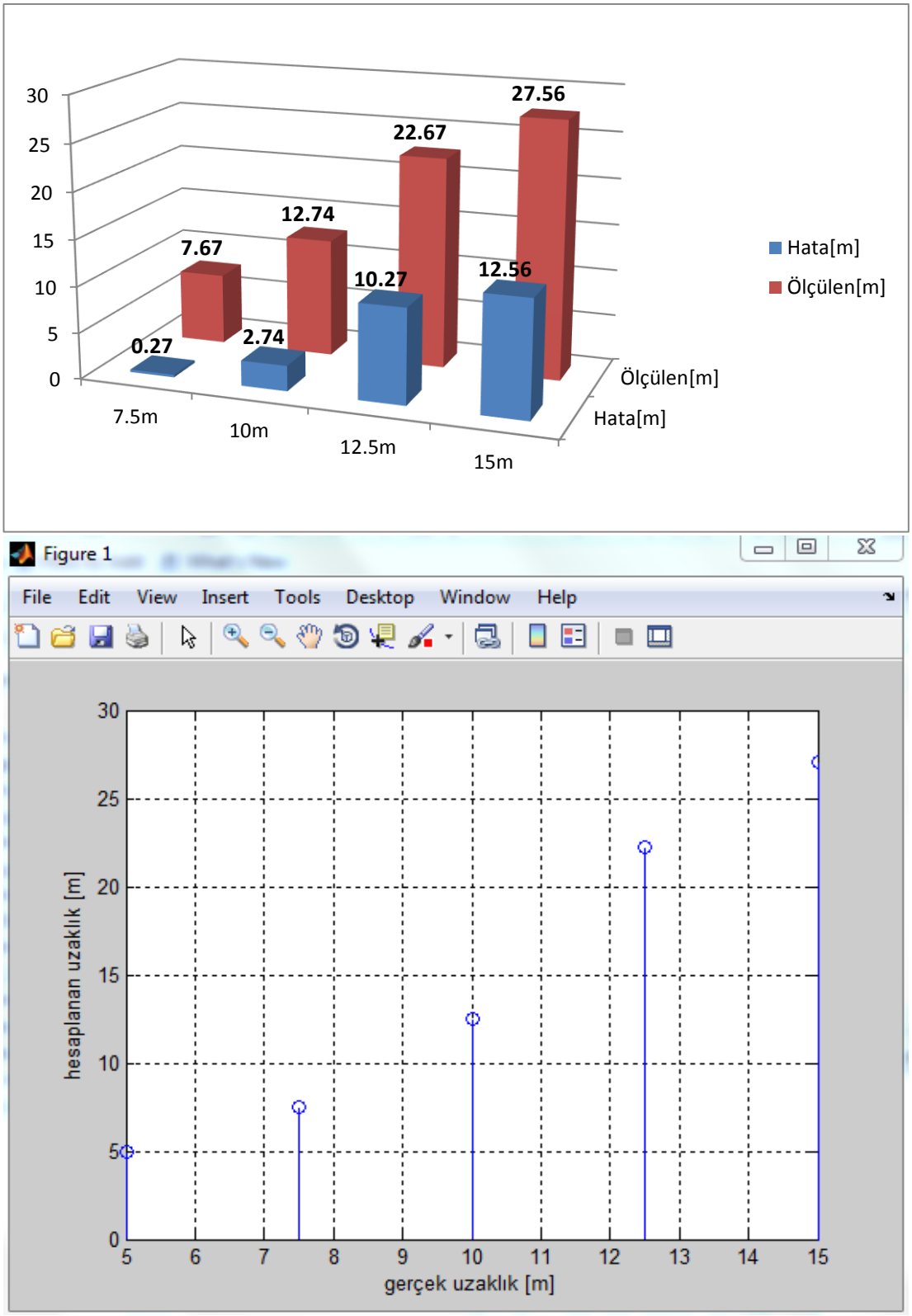


Şekil 3.32. Çanta için 15m.'deki görüntü

Görüntülerin HSV uzayında eşiklenmesi sonucunda yukarıdaki şekillerden de gözlenebileceği gibi çanta haricinde gürültülerde bulunmaktadır. Gürültülerin giderilmesi için morfolojik işlemlerin yanında 'Connected Component Labeling' yöntemi kullanılmıştır.

Tablo 3.8. Referans görüntü 5m. iken nesne uzaklık ölçümü

Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
7,5	20887	7,67
10	12582	12,74
12,5	7073	22,67
15	5820	27,56



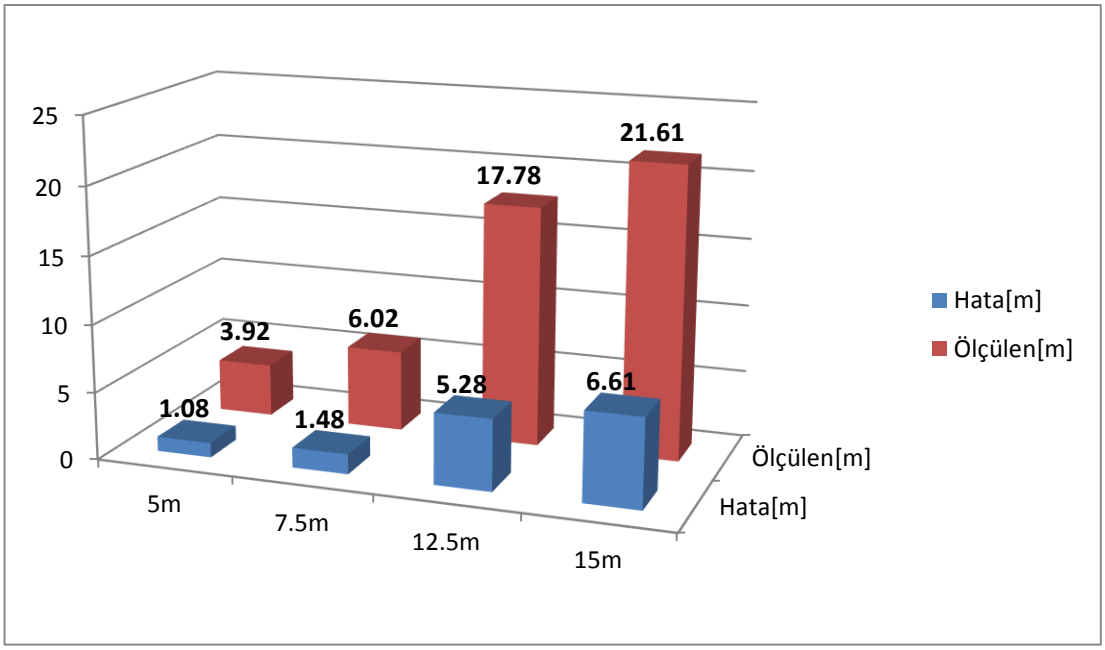
Şekil 3.33. Referans görüntü 5m. iken ölçüm sonuçları

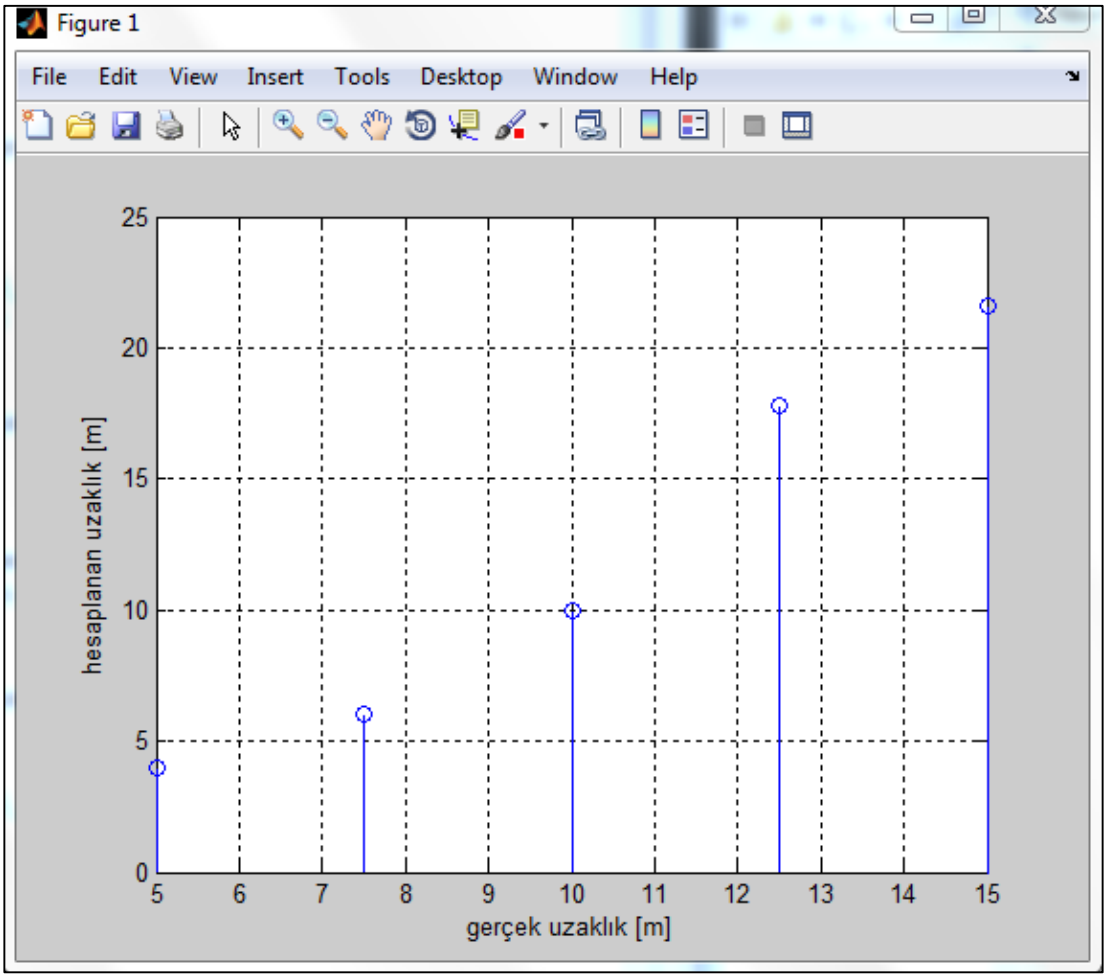
3.1.4.2. Kırmızı çanta referans görüntü 10m. iken ölçüm

Referans görüntü 10m. iken kırmızı çantanın kameraya olan uzaklığı ölçülmüştür. Ölçümler 5m., 7,5m., 12,5m. ve 15m. mesafe için yapılmıştır. Elde edilen sonuçlar Tablo 3.9'da gösterilmiştir. Şekil 3.34'te referans görüntü 10m. iken ölçülen uzaklıkların hata oranları gösterilmiştir.

Tablo 3.9. Referans görüntü 10m. iken nesne uzaklık ölçümü

Uzaklık [m.]	Piksel	Ölçülen uzaklık [m.]
5	32081	3,92
7,5	20887	6,02
12,5	7073	17,78
15	5820	21,61





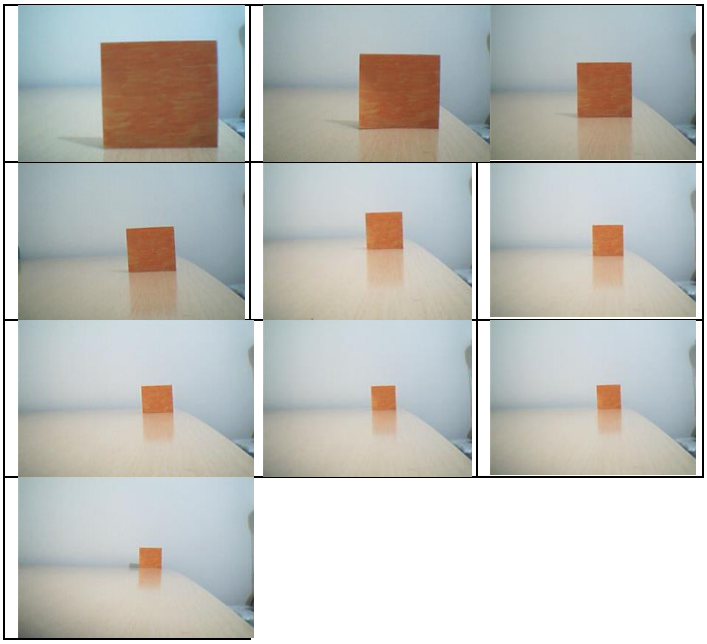
Şekil 3.34. Referans görüntü 10m. iken ölçüm sonuçları

Çalışmanın bu kısmına kadar, sadece tek referans görüntü kullanılarak nesne uzaklık hesabı yapılmıştır. Grafiklerden de gözleneceği gibi referans görüntüden uzaklaştıkça hata oranı artmaktadır. Bu sebeple birden fazla referans görüntü kullanılarak ölçüm yapılması düşünülmüştür. Yani görüntü işlemede sınıflandırma diye bilinen yöntem çalışmaya adapte edilmiştir. Alınan sonuçlar oldukça başarılıdır. Sınıflandırma kullanılarak yapılan ölçümler bir sonraki bölümde anlatılmaktadır.

3.2. Sınıflandırma Yöntemi Kullanılarak Yapılan Ölçümler

İlk olarak sistem için çalışma aralığı 20cm.-110cm. arası olarak belirlenmiştir. Çalışma aralığında 20cm.'den başlayarak her 10cm.'de bir görüntü alınarak sisteme tanıtılmıştır. Çünkü, nesne uzaklaştıkça piksel değişiminin cm türünden ifade edilişi değişmektedir. Böylece, her aralıktaki 1cm. değişimin o aralık için kaç pikselle ifade

edildiği belirlenmiştir. Şekil 3.35'te kullanılan referans nesnelerin görüntüleri gözükmektedir.



Şekil 3.35. 20cm.-110cm. arasında alınan nesnenin referans görüntüleri

20cm.-110cm. arası her 10cm.'de alınan görüntülerin sisteme tanıtılması, aşağıdaki gibi, Tablo 3.10'da gösterilmektedir.

Tablo 3.10. 20cm.-110cm. arası her 10cm.'de alınan görüntülerin sisteme tanıtılması

Referans Görüntü Uzaklıkları [cm.]	Referans Görüntüdeki Nesnenin Piksel Sayısı
20	30365
30	16166
40	9170
50	6621
60	4952
70	3825
80	3127
90	2592
100	2240
110	1960

Çalışma aralığının, 20cm.-110cm. olduğu düşünüldüğünde, 20cm. ve 110cm.'deki nesne piksel sayılarının farkının aradaki mesafe farkına oranı 1cm. değişimin kaç pikselle ifade edildiğini verecektir.

Yani;

30365-1960=28405 piksel

20cm.-110cm.=90cm.

$$\frac{28405 \text{ piksel}}{90 \text{ cm}} = 315{,}6 \left[\frac{\text{piksel}}{\text{cm}}\right]$$

Bu inceleme sonucunda şöyle bir sonuç çıkartılabilir;

Nesnenin kameradan 1cm. uzaklaşması veya yakınlaşması için görüntüde yaklaşık 315 piksellik bir değişim olması gerekmektedir.

Nesne 100cm.'de ve 110cm.'de iken piksel sayılarına bakıldığında

100cm. >> 2240 piksel

110cm. >> 1960 piksel

Nesne 100cm.'den 101cm.'e gittiğinde kapladığı piksel sayısı 315 azalmalıdır.

2240-315= 1915 piksel => 101cm. uzaklıktaki nesnenin kaplayacağı piksel sayısı

Referans olarak çekilen 110cm.'deki görüntünün piksel sayısı 1960'tır. Burada hesaplandığımız nesnenin 100cm.'den, 101cm.'e gitmesi için piksel sayısı 1915'tir.

Nesne piksel sayısı uzaklaştıkça piksel sayısı azalacağı için 1915 piksel sayısına sahip nesne 1960 piksel sayısına sahip nesneden daha uzaktaymış gibi hesaplanmaktadır.

Yani 90cm.'lik bir çalışma aralığının piksel değişimi, cm. cinsinden değişimi doğru vermemektedir.

20cm.-30cm. aralığında 1cm. için piksel değişimi ile 80cm.-90cm. aralığında 1cm. için piksel değişiminin aynı olmadığı belirlenmiştir. Bunun için her 10cm. aralıkta 1cm.'ye karşılık gelen piksel sayısı hesaplanmıştır. Şekil 3.36'da aralıklara göre, 1cm. değişimin piksel cinsinsen ifadesi gösterilmektedir.

Referans görüntü uzaklıkları	Referans görüntüdeki nesnenin piksel sayısı	
20 cm	30365 piksel	
30 cm	16166 piksel	Sınıf 1
40 cm	9710 piksel	Sınıf 2
50 cm	6621 piksel	Sınıf 3
60 cm	4952 piksel	Sınıf 4
70 cm	3825 piksel	Sınıf 5
80 cm	3127 piksel	Sınıf 6
90 cm	2592 piksel	Sınıf 7
100 cm	2240 piksel	Sınıf 8
110 cm	1960 piksel	Sınıf 9

Şekil 3.36. Oluşturulan sınıflar

Böylece her 10cm.'lik aralıkta 1cm.'lik değişim farkı, farklı piksel sayılarıyla ifade edilmiştir.

İyileştirilmiş algoritma:

1. Kameradan alınan görüntünün kaç piksel kapladığı hesaplanmaktadır.
2. Hesaplanan piksel sayısı referans olarak alınan 10 görüntünün piksel sayılarıyla karşılaştırılmaktadır.
3. Hesaplanan piksel hangi aralıktaysa nesnenin uzaklığı, o aralıktaki 1cm.'lik değişimin kaç pikselle ifade edildiği göz önünde bulundurularak hesaplanmaktadır.

Örneğin hesaplanan piksel değeri 10000 olsun;

- Hesaplanan piksel değeri, Referans görüntülerin piksel değerleriyle karşılaştırıldığında 30cm.-40cm. arasında olduğu belirlenmektedir.

30cm. ≥ 16166 piksel

40cm. ≥ 9710 piksel

- 30cm. ile 40cm. arasındaki 1cm.'lik değişim 645,6 piksele denk gelmektedir.

16166-9710=6456piksel ≥ 6456÷10=645,6 piksel/cm

oran=(pixel_30-pixel_40)/10;%pixel değişiminin cm.'ye göre ifadesi

oran=645,6[piksel/cm];

- Hesaplanan piksel değeri ile 40cm.'deki görüntünün piksel sayısına göre farkı bulunur.

pixel_farki=hesaplanan_pixel-pixel_40;

pixel_farki=10000-9710= 290[piksel];

- Piksel olarak elde edilen 40 cm.'e göre uzaklık cm. cinsine çevrilir.

cmuzaklik=pixel_farki/oran;

cmuzaklik=290/645,6= 0,4496 [cm.];

uzaklik=40-cmuzaklik

uzaklik=40-0,45=39,55 [cm.];

Bu şekilde sınıflandırma yöntemi kullanılarak gerçekten 40cm. olan uzaklık 39,55cm. olarak hesaplanmıştır.

20cm.-110cm. aralığında nesne uzaklığı yaklaşık 2-3cm. hata ile hesaplanabilmektedir.

3.3. Histogram Eşikleme Yöntemi Kullanılarak Yapılan Ölçüm

Bölüm 1'de histogram eşikleme yöntemi hakkında teorik bilgi verilmiştir. Bu kısımda yöntem kullanılarak mesafe ölçümü için yapılan sonuçlar gösterilmektedir. Mesafe ölçüm işlemi video üzerinde gerçek zamanlı olarak gerçekleştirilmiştir. İşlemler top ve silgi olmak üzere iki farklı nesne için yapılmıştır.

3.3.1. Top için histogram eşikleme kullanılarak yapılan mesafe ölçüm sonuçları

Kırmızı renkli top kullanılarak çekilen video üzerinden topun gerçek zamanlı hareketi incelenmiştir. İlk aşamada üzerinde çalışılan videodan referans olarak kullanılacak görüntü çekilmektedir. Videoda 1sn'de yaklaşık 30 görüntü alınmaktadır.

Videodan görüntü çekmek için kullanılan kod parçası aşağıdadır.

xyloObj = VideoReader('top_video.mp4');

```
firstFrame = read(xyloObj, 3);%referans görüntü
```

Bu çalışmada referans görüntü olarak 100cm. mesafedeki videonun 3. İmgesi kullanılmıştır. Her görüntü gri uzaya çekilerek histogram değerine göre nesneye ait gri seviyesine bakılmıştır. Bu yöntemle nesne gri seviyesindeki piksel sayısı belirlenmiştir. Aşağıdaki kod parçası nesneye ait piksel sayısını vermektedir.

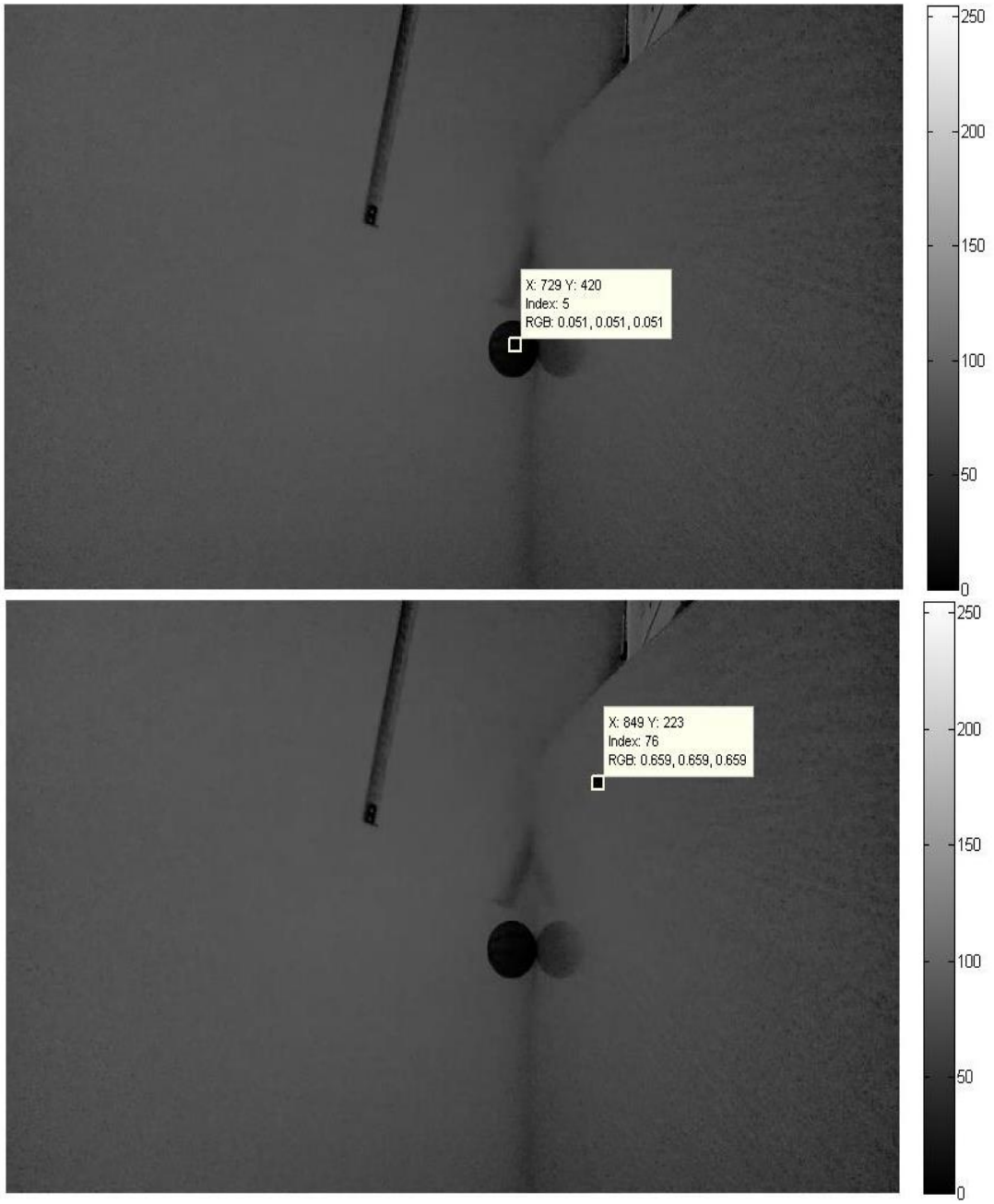
```
ref_gray_imge=rgb2gray(firstFrame);

bw = (ref_gray_imge<25); %nesnenin gri seviye eşiği

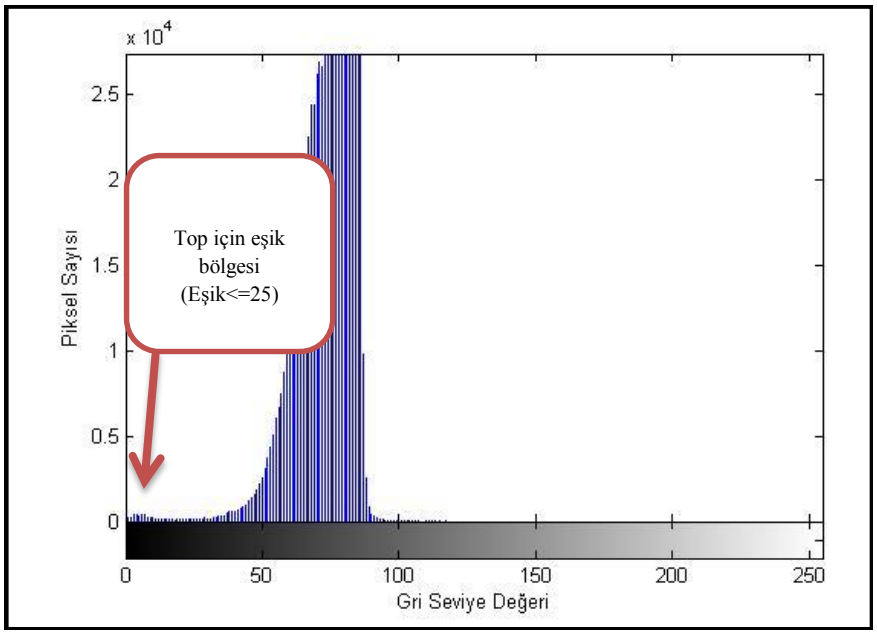
h_1=imhist(ref_gray_imge);

ref_pixel=mean(sum(h_1(1:25)));%referans görüntü için nesneye ait piksel sayısı
```

Şekil 3.37'de referans görüntü ve histogram grafiği gösterilmiştir.

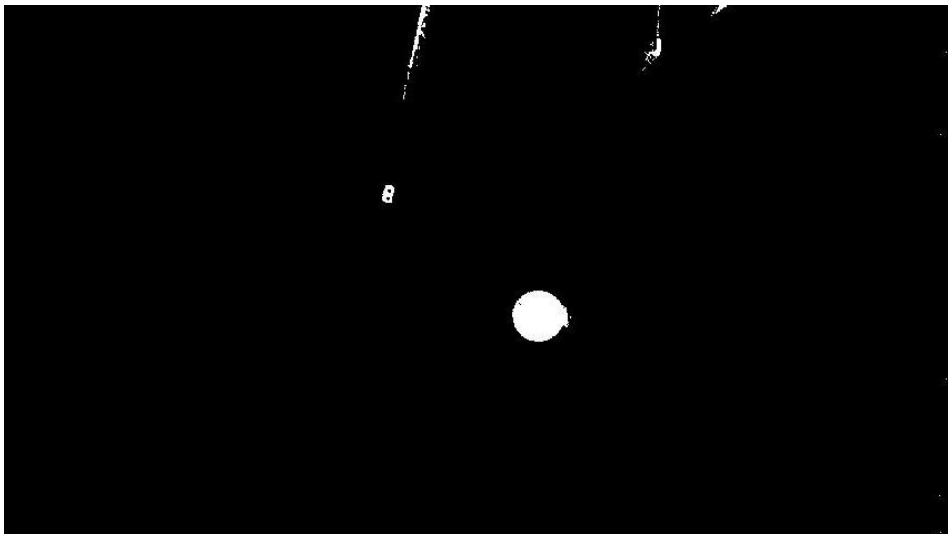


Şekil 3.37. Referans görüntü ve histogram grafiği



Şekil 3.37. (Devam) Referans görüntü ve histogram grafiği

Şekil 3.37'den gözleneceği gibi topun gri seviyesi 25'ten aşağıdadır. Eşikleme sonucunda elde edilen 100cm.'deki referans görüntü Şekil 3.38'de gösterilmektedir.



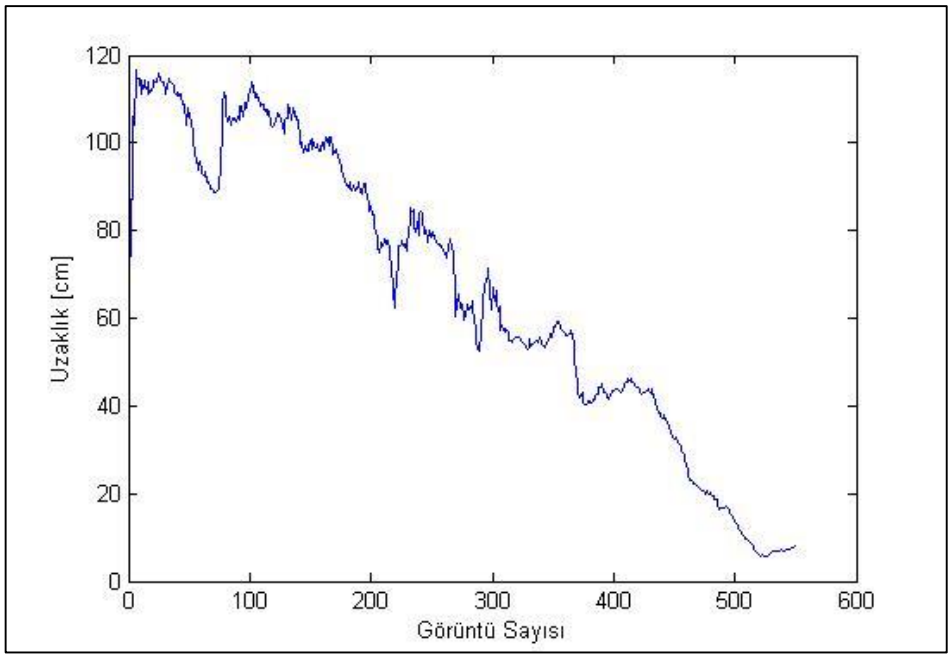
Şekil 3.38. 100cm.'deki top için referans görüntü

Histogram değerine göre 25'ten küçük olan gri seviyesindeki pikseller toplanmıştır. Aşağıdaki kod parçası histogramdan nesneye ait piksel sayısını vermektedir.

```
h_1=imhist(ref_gray_imge);

ref_pixel=mean(sum(h_1(1:25)));%referans görüntü için nesneye ait piksel sayısı
```

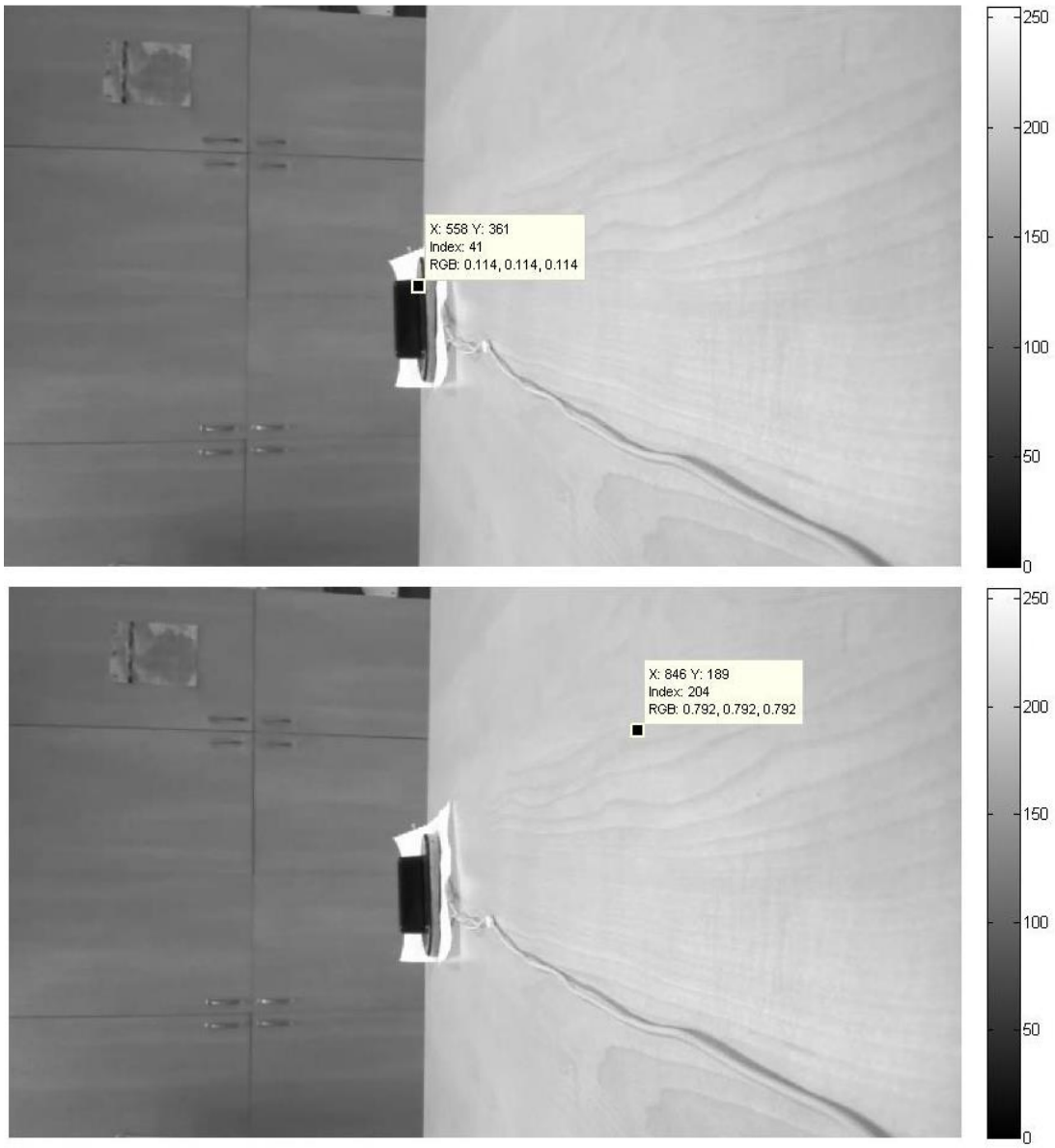
Video içindeki tüm görüntüler incelenerek yukarıdaki kod parçası yardımıyla nesne için piksel sayısı hesaplanmaktadır. Hesaplanan piksel ve referans piksel kullanılarak topun kameraya olan uzaklığı video üzerinden gerçek zamanlı olarak ölçülmüştür. Yaklaşık %20 hatalı çıkmaktadır. Yaklaşık 18sn'lik videodan nesnenin gerçek zamanlı mesafe ölçüm sonuçları Şekil 3.39'da gösterilmektedir.



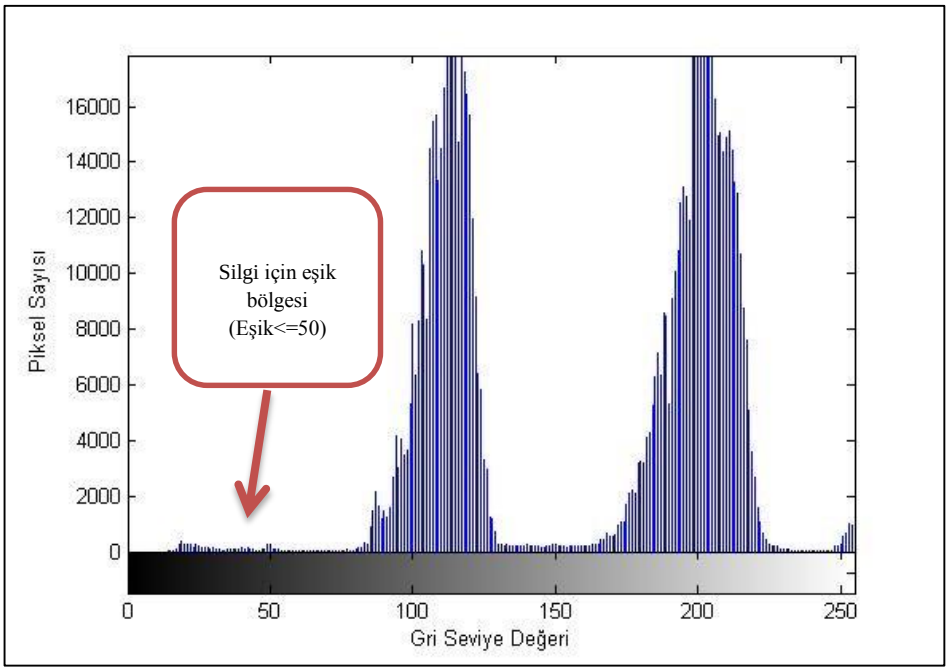
Şekil 3.39. Top için gerçek zamanlı mesafe ölçüm sonuçları

3.3.2. Silgi için histogram eşikleme kullanılarak yapılan mesafe ölçüm sonuçları

Siyah renkli silgi kullanılarak çekilen video üzerinden silginin gerçek zamanlı hareketi incelenmiştir. İlk aşamada üzerinde çalışılan videodan referans olarak kullanılacak görüntü çekilmektedir ve histogram grafiğine bakılarak silgi için uygun gri eşik seviyesi belirlenmiştir. Videodan referans ve diğer görüntüleri çekmek için kullanılan kodlar top için önceki bölümde verilmişti. Silgi için de aynı kodlar kullanılmıştır. Tek farklı nokta histogram eşikleme değeridir. Şekil 3.40'da silgi ve histogram eğrisi gösterilmektedir.

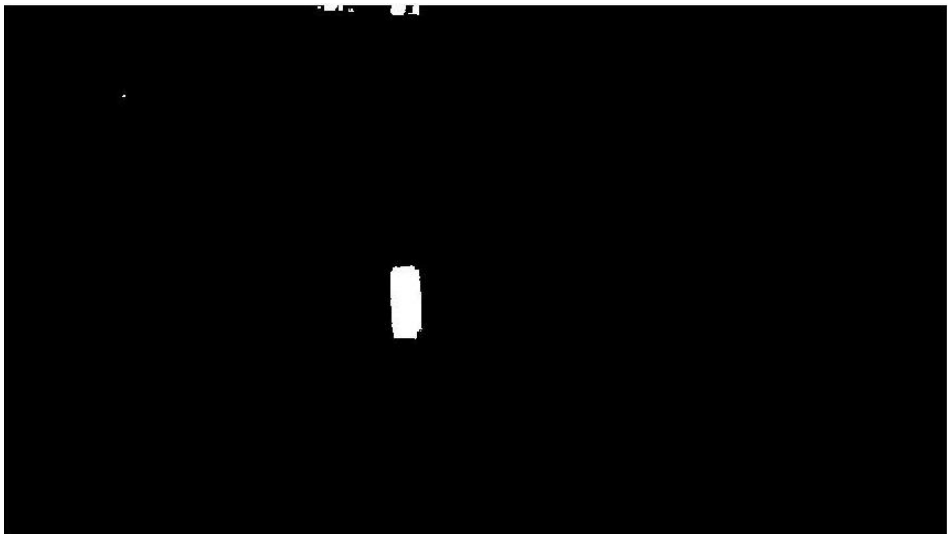


Şekil 3.40. Silgi için referans görüntü ve histogram eğrisi



Şekil 3.40. (Devam) Silgi için referans görüntü ve histogram eğrisi

Şekil 3.40'dan gözleneceği gibi silgi gri seviyesi 50'den aşağıdadır. Eşikleme sonucunda elde edilen 100cm.'deki referans görüntü Şekil 3.41'de gösterilmiştir.



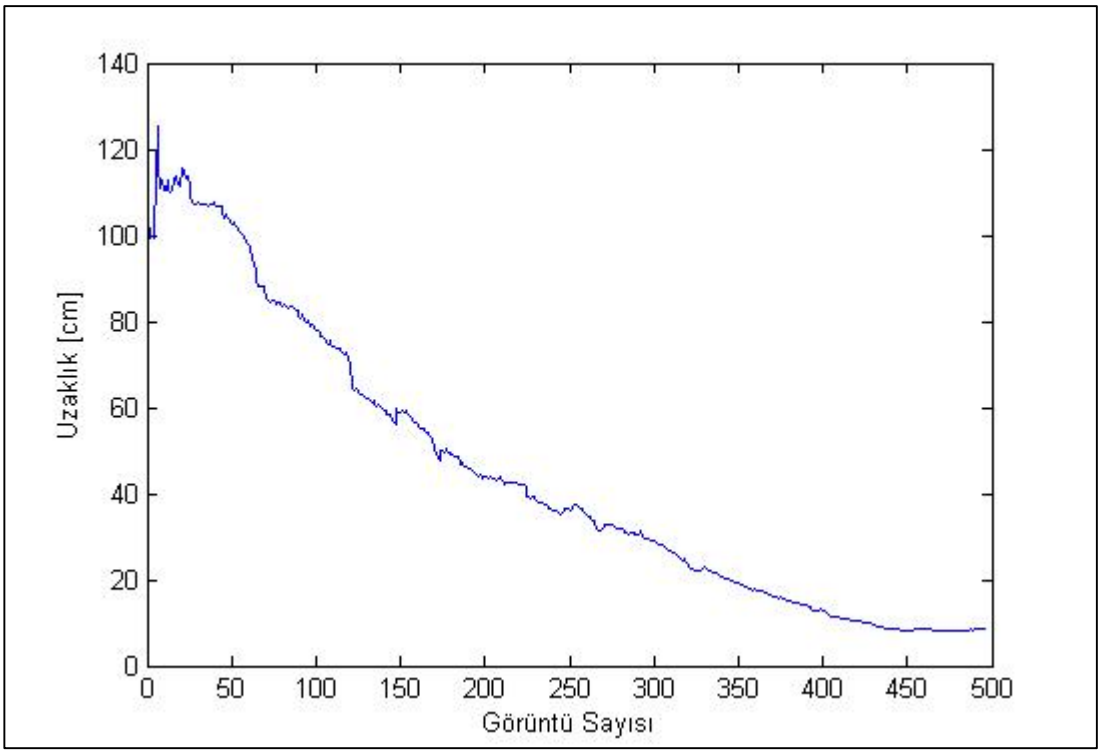
Şekil 3.41. 100cm.'deki silgi için referans görüntü

Histogram değerine göre 50'den küçük olan gri seviyesindeki pikseller toplanmıştır. Aşağıdaki kod parçası histogramdan nesneye ait piksel sayısını vermektedir.

```
h_1=imhist(ref_gray_imge);

ref_pixel=mean(sum(h_1(1:50)));%referans görüntü için nesneye ait piksel sayısı
```

Video içindeki tüm görüntüler incelenerek yukarıdaki kod parçası yardımıyla nesne için piksel sayısı hesaplanmaktadır. Hesaplanan piksel ve referans piksel kullanılarak silginin kameraya olan uzaklığı video üzerinden gerçek zamanlı ölçülmüştür. Yaklaşık %00 hatalı çıkmaktadır. Yaklaşık 18sn'lik videodan nesnenin gerçek zamanlı mesafe ölçüm sonuçları Şekil 3.42'de verilmiştir.



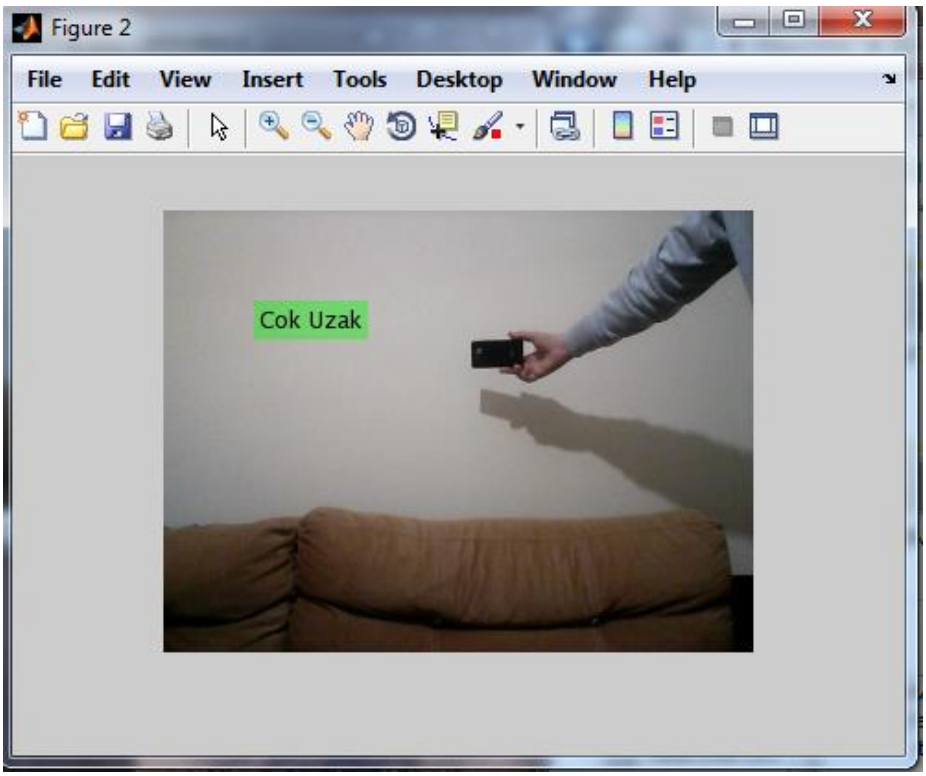
Şekil 3.42. Silgi için gerçek zamanlı mesafe ölçüm sonuçları

Topa göre silginin daha başarılı sonuç vermesi silginin daha net siyah renge sahip olmasından kaynaklanmaktadır. Top için uygun ışıklandırma ortamı oluşturulamamış olması hatalı sonuçlar alınmasının bir diğer sebebidir.

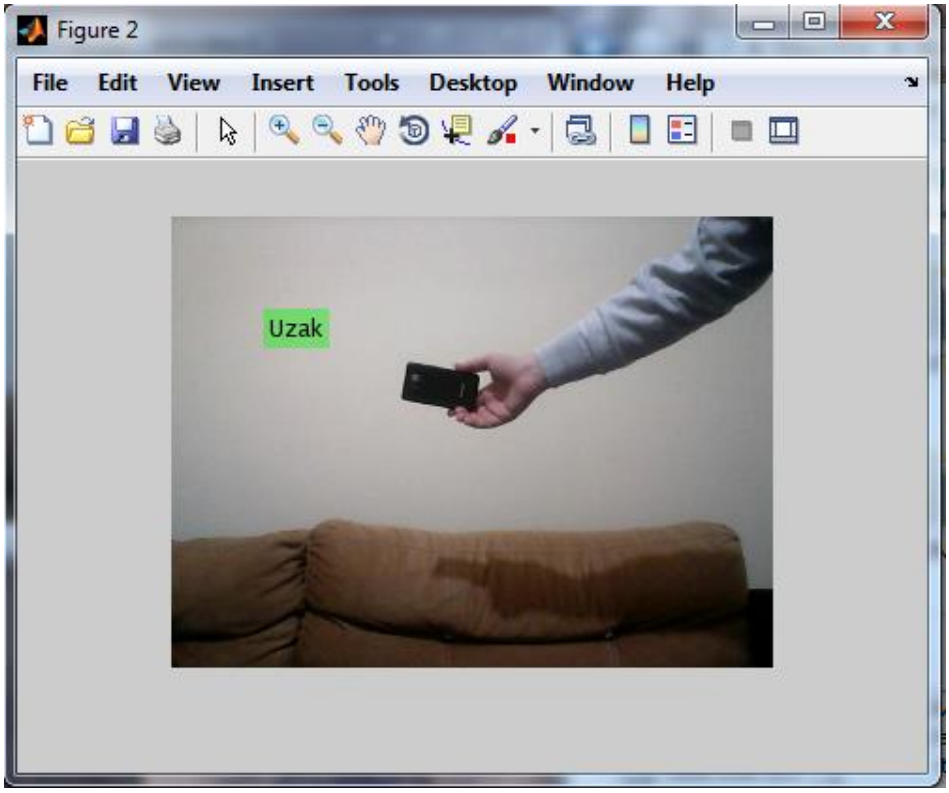
3.3.3. Gerçek zamanlı mesafe uyarım çalışması

Yapılan çalışmalar doğrultusunda, sistemin gerçek zamanlı olarak web kamera ile nesnenin kameraya yaklaşma ve uzaklaşma bilgisini verdiği belirlenmiştir. Sistem gri ton eşik değeri 40 ve altında olan cisimler üzerinde test edilmiştir. Kullanıcı tarafından istenen mesafe değişim oranı uyarı olarak gösterilmektedir. Bu çalışmanın temeli, nesne kameraya yaklaştıkça nesnenin gri tonundaki beyaz piksel sayısının artması, nesne kameradan uzaklaştıkça beyaz piksel sayısının azalması mantığına

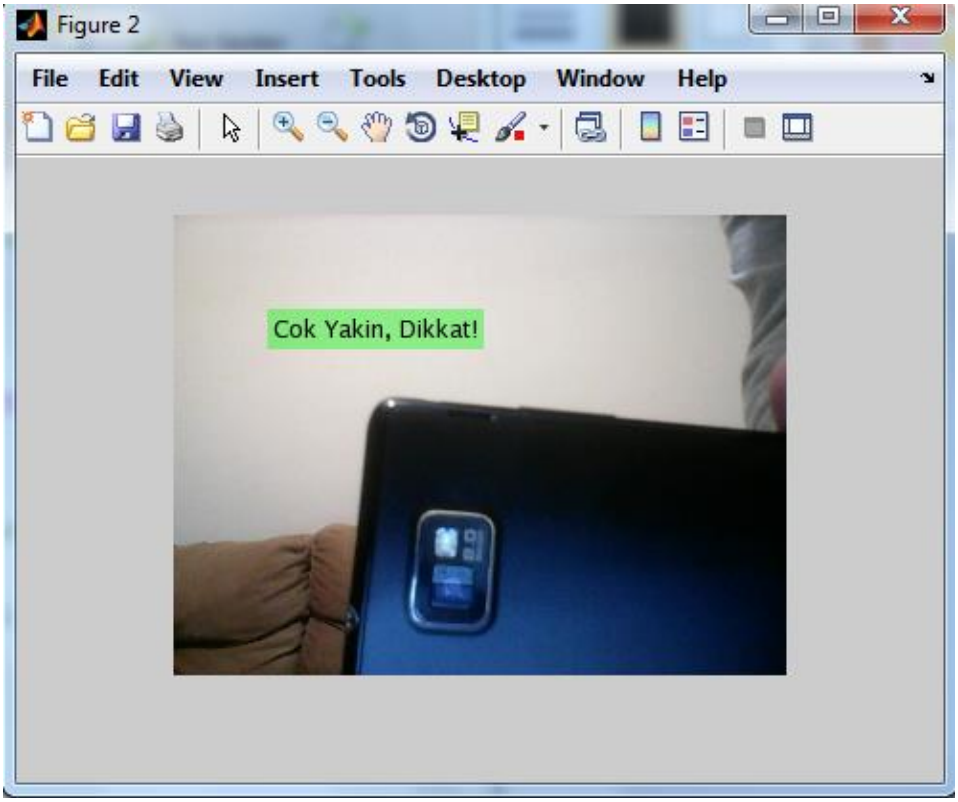
dayanmaktadır. Şekil 3.43, Şekil 3.44 ve Şekil 3.45'te kameraya yaklaşan nesnenin konumuna göre uzaklık bilgisinin verildiği görülmektedir.



Şekil 3.43. Gerçek zamanlı nesne uyarı bilgisi–'çok uzak'



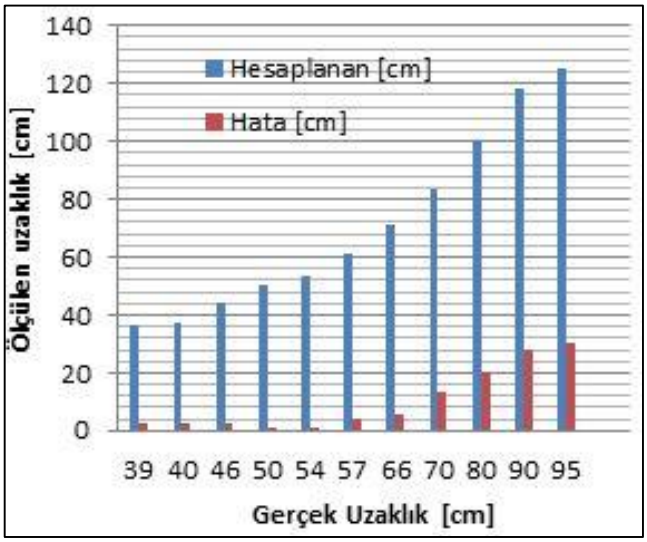
Şekil 3.44. Gerçek zamanlı nesne uyarı bilgisi–‘uzak’



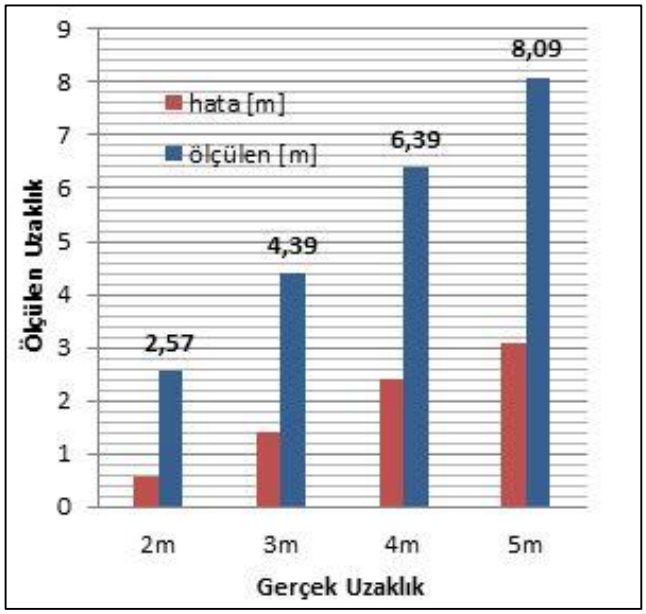
Şekil 3.45. Gerçek zamanlı nesne uyarı bilgisi–‘çok yakın!’

3.4. Başarım Oranları

Bu tez çalışmasında uygulanan görüntü işleme metotlarının deneysel sonuçlarına bu kısımda yer verilmiştir. Uygulanan farklı görüntü işleme algoritmalarının sonucunda elde edilen deneysel sonuçlar yorumlanmıştır. Alınan örnek sonuçların görüntüleri verilmiş olup bu görüntüler incelenerek başarım oranlarından bahsedilmiştir. Şekil 3.46'da 50cm. referans görüntü için, Şekil 3.47'de 1m. referans görüntü için ölçüm sonuçları görülmektedir.

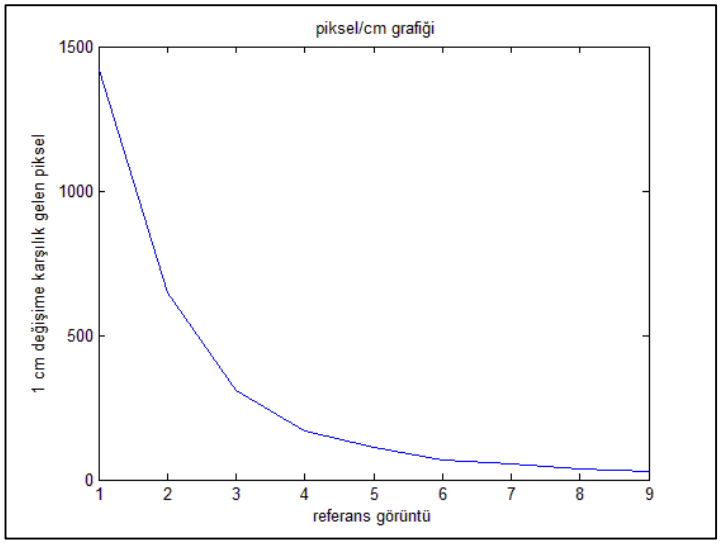


Şekil 3.46. 50cm. referans görüntüye göre ölçüm sonuçları

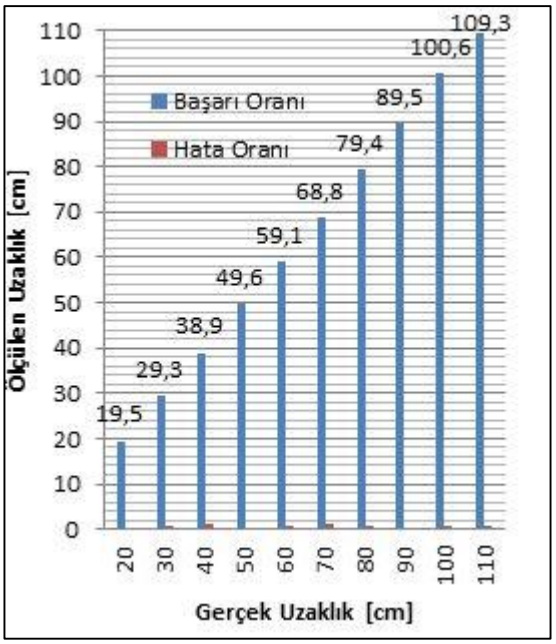


Şekil 3.47. 1m. referans görüntüye göre ölçüm sonuçları

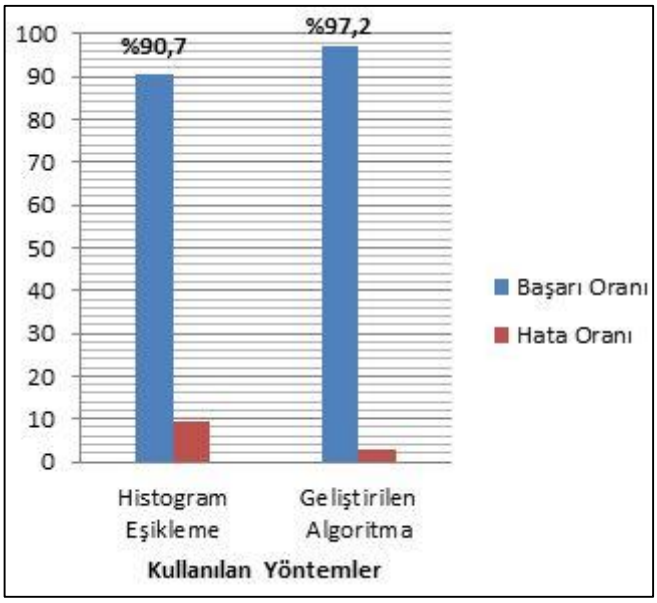
Şekil 3.48'de nesnenin kameradan uzaklaştıkça 1cm.'e denk gelen piksel sayısı gösterilmiştir. Şekil 3.49'da geliştirilen yöntemin sonuçları ve Şekil 3.50'de tez çalışmasında kullanılan yöntemlerin başarım oranları gösterilmiştir.



Şekil 3.48. Piksel/cm grafiği



Şekil 3.49. Geliştirilen yönteminin sonuçları



Şekil 3.50. Kullanılan yöntemlerin karşılaştırılması

4. SONUÇLAR VE ÖNERİLER

Nesnelerin uzaklığının belirlenmesinde farklı yöntemler kullanılmaktadır. Nesne uzaklığı belirlenmesinde tek kamera veya iki kamera kullanılabilmektedir. Kinect kameralar kullanılarak; görüntüden derinlik bilgisi elde edilebilmektedir. Kinect kamera iki adet kameradan oluşmaktadır.

Bu çalışmada, tek bir web kamerası kullanılarak renk tabanlı nesne takibi gerçekleştirilmiş ve nesne uzaklık ölçümünde kullanılabilecek bir yöntem geliştirilmiştir. Geliştirilen yöntemin nesne uzaklık ölçümündeki başarımı histogram eşikleme yöntemiyle karşılaştırılmıştır. Geliştirilen yöntemin başarım oranı % 97,2'dir. Sistemin cevap verme süresi yaklaşık olarak 95ms'dir. Yöntem 20cm.-110cm. arasında test edilmiştir. Daha geniş çalışma aralığında da başarım oranının çok değişmeyeceği düşünülmektedir, çünkü başarım oranı alınan referans görüntü sayısına bağlıdır. Ne kadar çok referans görüntü alınır ve buna bağlı olarak ne kadar çok sınıf oluşturulursa başarım oranı da o kadar artacaktır. Sistem kübik ve silindirik nesneler için gayet başarılı çalışmaktadır. Ayrıca nesnenin kameraya olan uzaklığını daha az hata ile belirleyebilmek için, deneysel gözlemler sonucu algoritmada iyileştirme gerçekleştirilmiştir. Ölçümün doğru yapılmasında, 1cm. yer değiştirmenin kaç piksellik bir artış veya azalışa karşılık geldiğinin belirlenmesi önemli rol oynamıştır. Çalışma aralığında, 1cm.'ye karşılık gelen piksel sayısını belirleyebilmek için 10 referans görüntü kullanılmıştır. Nesne kameradan uzaklaştıkça 1cm.'ye denk gelen piksel sayısının logaritmik olarak azaldığı gözlemlenmiştir.

Bu bilgiler ışığında iyileştirilen algoritmada nesnenin kameraya olan uzaklığı yaklaşık olarak %4 hata ile belirlenebilmiştir. Algoritma mobil robot uygulamalarında 0,2m.-5m. aralığında aynı hata oranıyla kullanılabilir. Çalışma aralığında daha referans görüntü ile hata payı azaltılabilir. Yine daha fazla referans görüntü kullanılarak çalışma alanı genişletilebilir. Ayrıca, referans görüntülerdeki nesne uzaklıkları yapay sinir ağlarında eğitilerek, uzaklık modeli çıkartılabilir.

KAYNAKLAR

[1] Canan S., Yapay sinir ağları ile GPS destekli navigasyon sistemi, Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2006, 183208.

[2] Yılmaz N., Sağıroğlu Ş. ve Bayrak, M., Genel Amaçlı Web Tabanlı Mobil Robot: SUNA, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 2006, **21**(4), 745-752.

[3] Yılmaz A., Kamera Kullanılarak Görüntü İşleme Yoluyla Gerçek Zamanlı Güvenlik Uygulaması, Yüksek Lisans Tezi, Haliç Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 200290.

[4] Melih Kuncan, H.Metin Ertunç, Gürkan Küçükyıldız, Berkan Hızarcı, Hasan Ocak, Sıtkı Öztürk, Görüntü İşleme Tabanlı Zeytin Ayıklama Makinesi, *Otomatik Kontrol Ulusal Toplantısı TOK2013*, Malatya, Türkiye, 26-28 Eylül 2013.

[5] Özden M., Ortalama Kayma Algoritmasının Geliştirilerek Görüntü Dizilerinde Hareketli Nesne Takibi ve Görüntü Kesimleme Amaçlı Kullanılması, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, Kırıkkale, 2005, 168786.

[6] Kuncan M., Endüstriyel bir makinada görüntü işleme yöntemi ile zeytinlerin renklerine göre ayırt edilmesi, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, 2013, 342536.

[7] Tonguç G., Görüntü İşleme Teknikleri Kullanılarak Meyve Tasnifi, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta, 2004, 200542.

[8] http://www.rapidtables.com/convert/color/rgb-to-hsv.htm, (Ziyaret tarihi:22 Haziran 2015).

[9] Günal S., Örüntü Tanıma Uygulamalarında Altuzay Analiziyle Öznitelik Seçimi ve Sınıflandırma, Doktora Tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2008, 178927.

[10] Güvenç H., Ortak Vektör Yöntemiyle Öznitelik Çıkarımı, Yüksek Lisans Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2009, 179112.

[11] Bellman R., *Adaptive Control Processes a Guided Tour*, 3th ed., Princeton University Press, New Jersey, 1961.

[12] Theodoridis S., Koutroumbas K., *Pattern recognition*, 1th ed., Academic Press, San Diego, 2006.

[13] Güral Şora E., Örüntü Tanıma Uygulamalarında Fraktal Boyut Yardımıyla Öznitelik Çıkarımı, Doktora Tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2010, 252663.

[14] Tunç S., Şekil Eşleştirmeye Dayalı Nesne Takibi Yöntemi Kullanılarak Video Kodlama, Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2010, 295604.

[15] Sayısal Görüntü İşleme Ders Notu, Prof. Dr. Bülent Bayram, YTÜ, http://www.yildiz.edu.tr/~bayram/sgi/saygi.htms, (Ziyaret tarihi: 30 Mart 2015).

[16] Florencio D. A. F., Schafer R. W., Decision-based median filter using local signal statistics, *Proc. of the SPIE Symposium on Visual Communications and Image Processing*, Chicago, USA, 16-18 September 1994.

[17] Non-Linear Median Filtering of Biomedical Image, Institute of Chemical Technology, Department of Computing and Control Engineering. http://dsp.vscht.cz/musoko/pdf/matlab02_mf.pdf, (Ziyaret tarihi: 5 Nisan 2015).

[18] Ayyüce Kızrak M., Özen F., A New Way of Looking of Fingerprint Recognition Median-Filtering Fingerprint Recognition Algortihm (HMFA), *Haberleşme Teknolojileri ve Uygulamaları Sempozyumu HABTEKUS 2009*, İstanbul, Türkiye, 9-11 Aralık 2009.

[19] http://www.atasoyweb.net/Component-Labeling, (Ziyaret Tarihi: 30 Mart 2015).

[20] http://www.johnloomis.org/ece563/notes/.html, (Ziyaret Tarihi: 30 Mart 2015).

[21] Gri Seviye Dönüşümleri ve Uzaysal Filtreleme BMÜ-357 Sayısal Görüntü İşleme Ders Notu, Fırat üniversitesi, Yrd. Doç. Dr. İlhan Aydın, http://web.firat.edu.tr/iaydin/bmu357/bmu_357_bolum2.pdf, (Ziyaret Tarihi: 30 Mart 2015).

EKLER

Ek-A

%Rgb görüntü okuma ve Hsv uzayına dönüştürme%

U_imge=imread('k_b2.jpg');% k_b2.jpg isimli rgb görüntüyü okuma

hsv_U_im=rgb2hsv(U_imge);% U_imge isimli rgb görüntüyü hsv formatına dönüştürme

%Belirlenen eşik aralığındaki tüm pikselleri dolaşma%

```
for i=1:1:1456%sütun sayısı

for j=1:1:2592%satır sayısı

if(hsv_U_im_r(j,i)>0.70 && hsv_U_im_r(j,i)<0.95)% nesne için örnek eşik

bos1(j,i)=hsv_U_im_r(j,i); % eşiğe uygun pikselleri boş bir alana al

end

end

end
```

% Genişletme ve Aşındırma işlemleri için matlab hazır fonksiyonları kullanılmıştır. Kullanılan fonksiyonlar aşağıda gösterilmektedir.%

```
%genişletme ve aşındırma için kullanılan matrisler

se = strel('disk',11);

se1 = strel('line',11,90);

bw3 = imdilate(bos1,se1);% imge genişletme

erodedBW1 = imerode(bw3,se);% imge aşındırma
```

% Referans olarak kabul edilen görüntüde nesne tespiti yapıldıktan sonra, nesnenin görüntüde kaç piksel kapladığı hesaplanmıştır. Nesnenin kaç pikselden oluştuğunu sayan kod parçası aşağıdadır. %

```
for i=1:1:1456

for j=1:1:2592

if(bw3(j,i)>0.7 && bw3(j,i)<0.95 )% belirlenen eşikteki piksel sayısı

pixel_Y=pixel_Y+1;

end
```

```
end

end

%Median Filtreleme%

med=medfilt2(hsv_U_im)

%Connected Component Labeling İşlemi %

function [piksel_sayisi]=CCL(BW)

[L,num] = bwlabel(BW);

M = im2uint8(L/num);

for i=1:num

area(i) = bwarea(L==i);

end

x=find(area>2500);

piksel_sayisi=area(x);

end

%Histogram Eşikleme%

imge=imread('ornek2.bmp'); % ornek2.bmp görüntüsünü okuma

gray_imge=rgb2gray(imge); % imge isimli görüntüyü rgb'den gray formata dönüştürme

h_1=imhist(gray_imge); % gray_imge görüntüsünün histogramını oluşturma

imhist(gray_imge);

figure

imshow(gray_imge)

%Gerçek zamanlı Video üzerinde mesafe bilgisi%

%kamera açılıyor

vid=videoinput('winvideo',1);

preview(vid);

pause(4);
```

```
i=1;

%Gerçek Zamanlı Çalışma%

clc;clear all; close all;

xyloObj = VideoReader('silgi.mp4');

numFrames = xyloObj.NumberOfFrames;

firstFrame = read(xyloObj, 3);%ilk frame üzerinde imcrop yapılır

ref_gray_imge=rgb2gray(firstFrame);

bw = (ref_gray_imge<50);% ref_gray_imge için eşik değer 50 olarak belirlenmiştir

h_1=imhist(ref_gray_imge);

ref_uzaklik=100;

ref_pixel=mean(sum(h_1(1:50)));% Eşik değeri 50'nin altında olan toplam piksel sayısı

position = [23 373];

box_color = {'red'};

writerObj = VideoWriter('uyarili_video');% uyarili_video üzerine yaz

writerObj.FrameRate = 30;

open(writerObj);

for i=1:numFrames

firstFrame1 = read(xyloObj, i);

ref_gray_imge=rgb2gray(firstFrame1);

bw = (ref_gray_imge<50);

h_1=imhist(ref_gray_imge);

hesaplanan_pixel=sum(h_1(1:50)); % Eşik değeri 50'nin altında olan piksel sayısı hesaplama

oran=ref_pixel/hesaplanan_pixel;

uzaklik(1,i)=oran*ref_uzaklik;

text_str = ['Yeterince Yakin! Uzaklik: ' num2str(uzaklik(i),'%0.2f') 'cm'];
```

```
if(uzaklik(1,i)<50)

firstFrame1 = insertText(firstFrame1,position,text_str,'FontSize',18,'BoxColor',box_color,'BoxOpacity',0.4);

end

writeVideo(writerObj,firstFrame1);

end

close(writerObj);

plot(uzaklik)
```

KİŞİSEL YAYIN VE ESERLER

[1] **Kuncan F.**, Yıldırım M., Tek Kamera Kullanarak Nesne-Kamera Arasındaki Uzaklığın Belirlenmesi İçin Bir Yöntem, *Otomatik Kontrol Ulusal Toplantısı TOK-2015*, Denizli, Türkiye, 10-12 Eylül 2015.

[2] Çubukçu A., Öztürk S., Kuncan M., Kaplan K., **Kuncan F.**, The İmplementation of DC Motor Speed Control Using Image Processing and OPC Interface in the S7-1200, *3rd Internatıonal Symposıum On Innovatıve Technologıes In Engıneerıng And Scıence Isıtes 2015*, Valencia, Spain, 3-5 June 2015.

ÖZGEÇMİŞ

Fatma Kuncan Eylül 1987, Batman doğumludur. İlk, orta ve lise öğrenimini Batman'da tamamlamıştır. 2005 yılında kazandığı Harran Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2009 yılında mezun olmuştur. Şubat 2011 tarihinde Siirt Üniversitesi Bilgisayar Mühendisliği Bölümü'nde ÖYP araştırma görevlisi olarak göreve başlamıştır. Evli ve 1 çocuk annesidir.